DLA-94-P20250

AUTOMATED BEST VALUE MODEL DECISION SUPPORT SYSTEM

FUNCTIONAL DESCRIPTION

DECEMBER 1993

FOR

DEPARTMENT OF DEFENSE
DEFENSE LOGISTICS AGENCY
Director (Procurement)
CAMERON STATION
ALEXANDRIA, VA 22304-6100

INSIGHT THROUGH ANALYSIS

DORO CORPORATE RESEARCH

DLA-94-P20042

AUTOMATED BEST VALUE MODEL DECISION SUPPORT SYSTEM

FUNCTIONAL DESCRIPTION

DECEMBER 1993

DEPARTMENT OF DEFENSE
DEFENSE LOGISTICS AGENCY
Executive Director (Plans & Policy Integration)
CAMERON STATION
ALEXANDRIA, VA 22304-6100

TABLE OF CONTENTS

LIST OF FIGURES

FUNCTIONAL DESCRIPTION
FOR
AUTOMATED BEST VALUE MODEL
DECISION SUPPORT SYSTEM

SECTION 1. GENERAL

1.1 Purpose of the Functional Description.

This Functional Description for the Automated Best Value Model (ABVM) Decision Support System (DSS) is written to provide:

a. The system requirements to be satisfied which will serve as a basis for mutual understanding between the user and the developer.

b. Information on performance requirements, preliminary design considerations, and user impacts including fixed and continuing costs.

c. A preliminary basis for development of system tests.

1.2 Project References.

This effort is an extension of the development of the ABVM (formerly the Defense Logistics Agency (DLA) Vendor Rating System or DVRS) under DLA Operations Research Office (DORO) projects DLA-92-P10164, DLA-94-P20249 and DLA-94-P20250. This effort will lead to the development of management information software. The sponsor for this project is the Headquarters, DLA Directorate of Procurement (AQP). This system will be utilized by procurement personnel at DLA's supply centers.

Applicable reference documents include:

a. Paul Grover, Randal Wendell, Major Mark Melius and Donna Smith, "Defense Logistics Agency Vendor Rating System," DLA-92-P10164, DLA Operations Research and Economic Analysis Office, September 1992.

b. Paul Grover, Randal Wendell, Major Mark Melius and Donna Smith, "Defense Logistics Agency Vendor Rating System - Technical Report," DLA-92-P10164, DLA Operations Research and Economic Analysis Office, August 1992 (unpublished draft).

c. Analytic Services Agreement, Automated Best Value Model (ABVM) Decision Support System, DLA-XX-P20250, March 1993. (Appendix A)

d. DLA Operations Research Office letter, Subject: Revised Automated Best Value Model (ABVM) Study Advisory Group (SAG) Meeting Minutes, 13 April 1993.

e. DLA Operations Research Office letter, Subject: Automated Best Value Model (ABVM) Decision Support System Report Formats (Project Number, DLA-XX-20250), 15 June 1993.

f. DoD STD 7935A, Military Standard DoD Automated Information Systems (AIS) Documentation Standards, 31 October 1988.

1.3 Terms and Abbreviations.

Acronym	Definition
ABVM	Automated Best Value Model
DLA	Defense Logistics Agency
DORO	DLA Operations Research Office
DPACS	DLA Preaward Contracting System
DSS	Decision Support System
SAMMS	Standard Automated Materiel Management System

SECTION 2. SYSTEM SUMMARY

2.1 Background.

ABVM has been developed as a means of assisting DLA contracting officers in making awards on the basis of the best overall value to the Government, rather than strictly on the lowest bid price. ABVM is designed to be implemented as part of DPACS at several of DLA's Supply Centers.

The proposed system is a DSS which can be used to provide management reports and to assess the benefits achieved through the implementation of ABVM. It is likely that this DSS will interface primarily with DPACS but may also be required to interface with SAMMS to obtain some of the required data.

2.2 Objectives.

The proposed system will provide the following capabilities:

2.2.1 The ability to generate standard management reports for evaluating the implementation of ABVM. Standard reports may be generated on a periodic basis (e.g., monthly), or may be generated in an on-line session.

2.2.2 An ad hoc query capability for non-standard reports. This capability will allow on-line access to identified data elements for formulation of non-standard queries.

2.3 Existing Methods and Procedures.

There is currently no method for examining DLA's ABVM program. Current management reports were developed prior to the ABVM program, and do not provide appropriate or sufficient information to evaluate the ABVM program. Figure 1 provides an extremely simplified generic overview of the existing procurement information flow utilizing the ABVM program. A purchase request identifies the requirement to obtain materiel, the procurement office prepares a solicitation which is made available to

private contractors, interested vendors submit bids to the procurement office describing the price and terms they are offering, the procurement office considers the bids along with vendors' performance history to evaluate the bids, and an award is made to the bidder representing the best value to the government.

The DPACS is used by DLA's procurement offices to assist in the automation of general procurement processes. Additionally, some of the historical award information within DLA is maintained in SAMMS. Because the ABVM program is in its infancy, evaluating its implementation currently would entail writing special programs to extract appropriate information from DPACS and SAMMS.

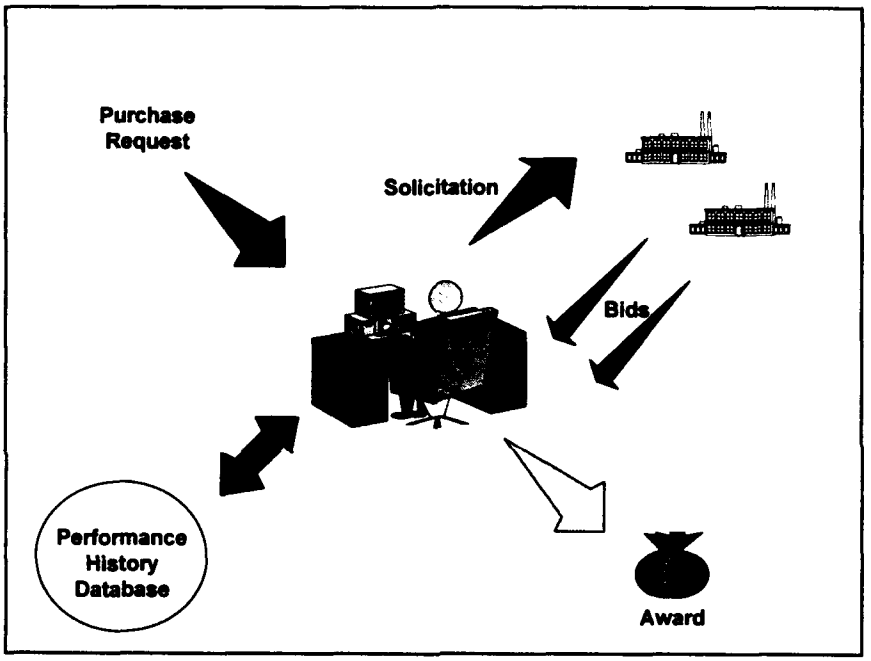


Figure 1 - Overview of Procurement Information Flow

2.4 Proposed Methods and Procedures.

The ABVM DSS will provide a new capability for assessing the implementation of the ABVM program. Management will be able to access the ABVM DSS independent of existing systems (unless the system is designed to directly access current production databases). The relationship of the ABVM DSS to existing programs and processes is depicted in Figure 2.

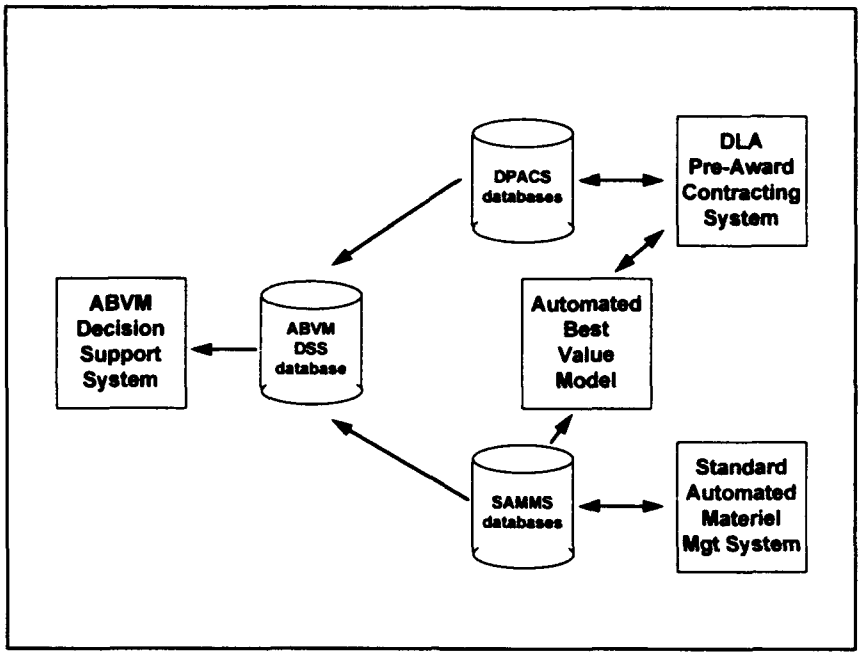


Figure 2 - ABVM DSS Relationship to Existing Processes

2.4.1 Summary of Improvements.

The ABVM DSS does not directly support DLA's procurement processes (other than in an informational manner), therefore, it will not in itself directly improve these processes. However, since the ABVM DSS will allow evaluation of ABVM program implementation, it may identify areas where the implementation should be adjusted to yield greater benefits. Insight gained from use of the ABVM DSS may also yield improvements in other areas as well.

2.4.2 Summary of Impacts.

2.4.2.1 User Organizational Impacts.

The DLA user organizations will include AQP and the DLA Supply Centers. These users are not anticipated to have any significant user organizational impacts except, perhaps, for the designation of a user point of contact to serve as liaison with the computer operating center on issues related to this system.

Because this system is intended to utilize currently existing databases, no additional support personnel are envisioned to be required. Since this is not a mission critical system, no contingency operations will be required.

2.4.2.2 User Operational Impacts.

The user's interface with the computer operating center will primarily be with regard to obtaining technical assistance in the development of ad hoc reports, since the user will likely not have sufficient technical expertise to utilize these capabilities of the system without help. Otherwise, this will merely be another system to be supported. Since all data will be obtained from other systems, no additional burden is expected for the user.

2.4.2.3 User Development Impacts.

2.4.2.3.1 Training. Training will be required during development to familiarize the user community with operation of the system.

2.4.2.3.2 Testing. It is assumed that various levels of testing will occur. Users will be required to support testing through initial operating capability.

2.5 Assumptions and Constraints.

2.5.1 Availability of Valid Data.

The value of this DSS is linked to the availability of valid data. For this effort, several DPACS database schemata have been used to identify sources of input data. It has been assumed that valid data is being maintained for all data elements (despite some indications to the contrary). It is beyond the scope of this effort to judge the validity of all data elements being utilized. In those cases where valid data is not available from the specified sources, alternative sources for these data elements must be identified, or the reports must be modified to address this lack of available data.

2.5.2 Operational System Functionality.

The ABVM DSS will not require substantial changes in functionality for the operational systems which provide data to it. Minor changes such as the collection of additional data elements will likely be required.

2.5.3 Evolving Data Requirements.

Data requirements will likely evolve as users become familiar with the capabilities of the system. These changing data requirements may also lead to the requirement for new report formats.

SECTION 3. DETAILED CHARACTERISTICS

3.1 Specific Performance Requirements.

The system will provide two major capabilities: (1) producing pre-defined standard reports through either batch or interactive processing; and (2) processing ad hoc queries in an interactive environment.

3.1.1 Accuracy and Validity.

Accuracy and validity of the proposed data will be dependent upon the data extracted from the source systems (e.g., DPACS, SAMMS).

3.1.2 Timing.

3.1.2.1 Response time from receipt of input data to availability of system products: Collection of data from source systems may take overnight or longer depending on the availability of the source systems. Once the data has been collected, some time may be required for reformatting into an internal database format.

3.1.2.2 Response time to queries and updates: The system will not update data. Response time to queries may vary. Response time to queries for standard reports should be immediate (less than 10 seconds). Response time to ad hoc queries should be less than 5 minutes.

3.1.2.3 Sequential relationship of functions: None.

3.1.2.4 Priorities imposed by types of inputs and changes in modes of operation: The ABVM DSS is anticipated to operate in the same mode.

3.1.2.5 Any deviations from specified response times for peak load periods or contingency operations, as applicable: None.

3.1.3 Capacity Limits.

The ABVM DSS is designed to provide management review of the effects of ABVM implementation policy, therefore, access to the ABVM DSS may be somewhat limited. It appears likely that no more than 20 users at any supply center would require access to the ABVM DSS, although local management may desire wider access to the system.

As the ABVM DSS relies on data from other sources, storage requirements will be determined by the amount of data extracted from these other sources. If the ABVM DSS is designed to directly access these source systems, there would be no additional storage requirements, unless a greater than normal amount of data would require to be archived. It is desired that at least 1 year's worth of data be available for analysis of trends.

3.2 Functional Area System Functions.

3.2.1 Standard Reports.

The ABVM DSS will have the capability to produce management reports using standard formats. It is possible that the system might be designed to allow both interactive and batch requests for generation of these standard reports.

3.2.2 Ad Hoc Queries.

This decision support system will also have the capability to perform ad hoc queries to allow examination of ABVM related data in other than the pre-defined report formats. The ad hoc query subsystem will allow access to all of the data elements utilized in the standard reports. Because of the specialized nature of the ad hoc query subsystem, some technical expertise may be required for its use.

3.3 Inputs and Outputs.

3.3.1 Inputs.

Most information input to the decision support system is expected to be available in existing systems, however, a few new data elements will be required to be collected. Appendix B is a list of data elements ordered by source; it also identifies the reports (described below) in which they are used. The two prime sources of data for the DSS are DPACS and SAMMS although the collection of a few new data elements will be required. Appendix C provides a list of new data element requirements by report. Current data elements which will require archival are identified in Appendix D.

3.3.2 Outputs.

The ABVM DSS will provide output in 11 standard report formats. Desired report formats for each of the 11 standard reports are provided in Appendix E. Outputs from the ad hoc query subsystem may be any logical combination of available input data elements. The pre-defined standard reports are described below:

(Report 1) Summary ABVM Application Statistics - Provides, by Center, an overall summary describing the application of ABVM by Federal Supply Class (FSC).

(Report 2) Monthly ABVM Component Score Statistics - Provides a breakout of ABVM component scores and Center average scores for the population of bids, awards, and vendors.

(Report 3) Quality Vendor Program Information - Provides overview statistics for vendors that meet Quality Vendor Program criteria.

(Report 4) Center ABVM Implementation "Benefits" Indicators - Provides an overview of changes to key indicators before and after implementation of the ABVM program. This report will serve as the baseline and primary source for validation of ABVM benefits.

(Report 5) Challenge Statistics - Provides an overview of the processing of vendor challenges to negative line information.

(Report 6) ABVM Buyer Performance - Provides a plot describing buyer trends in making ABVM awards where each award is represented by a point. Points closer to the origin (minimal price differential and performance loss) are typically indicative of "best" buys, although other circumstances may prevail.

(Report 7) ABVM Program Award Performance - Provides visibility of performance of key indicators for awards made under the ABVM program and for awards made without ABVM.

(Report 8) Center ABVM Statistics - Provides an overview of ABVM award statistics with respect to socio-economic status of awardees.

(Report 9) Center ABVM Statistics by Solicitation Consideration Factors - Provides an overview of ABVM award statistics with respect to solicitation consideration factors (ABVM weights) and socio-economic status of awardees.

(Report 10) ABVM Score Trends - Provides graphical presentation of ABVM scoring trends for the population of vendors by Center.

(Report 11) Near-term deliveries - ABVM Differential Awards - Lists individual ABVM differential contracts with anticipated deliveries so that they may be more closely monitored.

3.4 Database/Data Bank Characteristics.

Appendix B provides a listing of data elements used in this system.

3.5 Failure Contingencies.

There are no failure contingencies for the ABVM DSS because it is not a mission critical system. Should the ABVM DSS fail, it can be "refreshed" at the earliest convenience utilizing the source databases.

SECTION 4. DESIGN CONSIDERATIONS

4.1 System Description.

Figure 3 provides an overview of the ABVM DSS and its interface to existing systems.

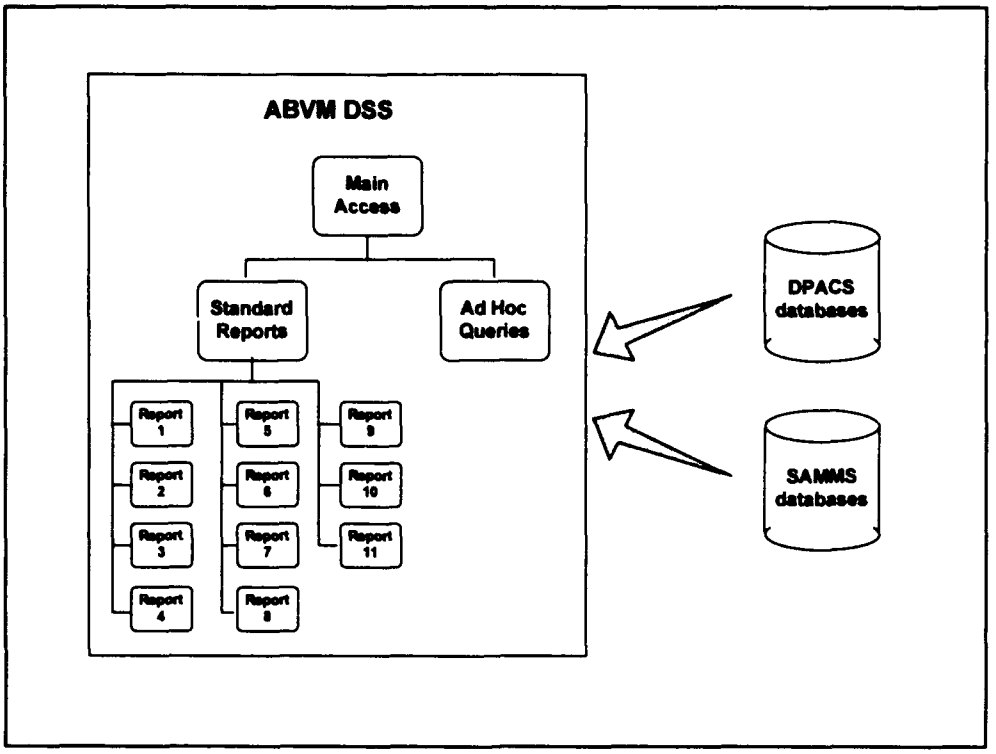


Figure 3 - Overview of ABVM DSS

4.2 System Functions.

Refer to Appendix F for a description of the computations required to produce the standard ABVM DSS reports.

4.3 Flexibility.

This system is a decision support system allowing management to evaluate the implementation of a new program - namely ABVM. As such, it is difficult to predict all of the potential ways that examination of ABVM implementation will be desired. Therefore, the design of this system should be sufficiently flexible to allow for changing management requirements such as the incorporation of additional standard reports. As future reports may utilize different data elements, the design should be able to accommodate the use of these additional data elements as well.

4.4 System Data.

There are no known special design requirements related to system data. As stated previously, this system is dependent upon data residing in several other existing databases. Depending on the environment chosen for development and operation of this system, it may be more practical to create a separate database for this system with periodic updates from the other databases. However, other considerations may dictate a system design which utilizes the existing databases more directly.

SECTION 5. ENVIRONMENT

To be determined (TBD). [Sections 5 through 8 should be written by system development personnel as they require specific technical expertise.]

SECTION 6. SECURITY

TBD

SECTION 7. SYSTEM DEVELOPMENT PLAN

TBD

SECTION 8. COST CONSIDERATIONS

TBD

APPENDIX A
ANALYTIC SERVICES AGREEMENT

ANALYTIC SERVICES AGREEMENT
AUTOMATED BEST VALUE MODEL (ABVM)
DECISION SUPPORT SYSTEM
DLA-XX-P20250

SECTION 1 - STUDY DESCRIPTION

1.1 **PROBLEM STATEMENT.** DLA is implementing an automated system to assist buyers in evaluating historical quality and delivery performance in making awards. There exists a need to examine/monitor the effectiveness of this system.

1.2 **OBJECTIVES.**

1.2.1 Determine appropriate management evaluation reporting requirements for validation of ABVM benefits.
1.2.2 Develop prototype reporting system.

1.3 **BACKGROUND.** Project DLA-XX-P20250 is a follow-on to project DLA-92-P10164.

1.4 **SCOPE.**

1.4.1 The ABVM prototype will not be altered.
1.4.2 This project is limited to the development of a functional description for a decision support system and a prototype of this system.
1.4.2 Only data elements presently available within DLA standard systems will be utilized.

1.5 **MAJOR ASSUMPTIONS AND CONSTRAINTS.**

1.5.1 Sub-indicators used within ABVM are valid measures of effectiveness.
1.5.2 Data required for evaluation of ABVM is available.

SECTION 2 - STUDY APPROACH

2.1 **ANALYTICAL TECHNIQUES APPLIED.** DLA-DORO will use applicable descriptive and inferential analysis methods in the design of the ABVM Decision Support System.

2.2 **SPECIAL REQUIREMENTS.**

2.2.1 Documentation provided by DORO as the functional description of the prototype decision support system will not fulfill all requirements of DoD Instruction 7935.1 AIS Documentation Standards. Further documentation may be required by a central design activity at the time of implementation.
2.2.2 The DLA hardware centers will provide advisors to consult on center requirements for management reporting related to ABVM.

2.2.3 The prototype decision support system will be off-line from the current DLA standardized systems such as SAMMS or DPACS, although it will utilize data generated from these systems.

2.2.4 Implementation of this prototype decision support system will be accomplished in a follow-on effort.

SECTION 3 - DELIVERABLES

3.1 Monthly progress reports.
3.2 Prototype reports.
3.3 Final briefing.
3.4 Functional description for decision support system.

SECTION 4 - MILESTONES

		Organization Primarily Responsible	
		LO/DORO	DLA-PP
4.1	ASA approval	12 Mar 93	
4.2	Meeting with center representatives to identify requirements.	30 Mar 93	30 Mar 93
4.3	Reporting requirements finalized.	12 May 93	12 May 93
4.4	Draft formats developed.	15 Jun 93	
4.5	Prototype methodology complete.	11 Aug 93	
4.6	Final prototype reports.	1 Sep 93	
4.7	Final briefing	15 Sep 93	
4.8	Functional description drafted.	13 Oct 93	
4.9	Functional description finalized.	10 Nov 93	
4.10	Anticipated Level-of-Effort:	750 hours equating to a cost of approximately $52,500.	

SECTION 5 - STUDY MANAGEMENT

5.1 **SPONSORING ORGANIZATION POCs.**

DLA-PPR Contact: Ms. Catherine Heretick
Phone: (DSN) 284-6431
FAX: (DSN) 284-0310

DLA-PS Contact: Mr. Phil Church
Phone: (DSN) 284-7866
FAX: (DSN) 284-0310

5.2 PERFORMING ORGANIZATION POCs.

DLA-LO	Contact:	Mr. Jim Russell Senior Study Director, Acquisition Management
	Phone:	(DSN) 284-7227
	FAX:	(DSN) 284-3831

DLA-DORO	Contact:	Mr. Kurt Schwarz Lead Analyst
	Phone:	(DSN) 695-5262
	FAX:	(DSN) 695-5319

5.3 OTHER ORGANIZATION POCs.

DSAC-OF	Contact:	Mr. Bill Eble
	Phone:	(DSN) 850-9707

ANALYTIC SERVICES AGREEMENT
AUTOMATED BEST VALUE MODEL (ABVM)
DECISION SUPPORT SYSTEM
DLA-XX-P20250

Christine L. Gallo
CHRISTINE L. GALLO
Executive Director
(Plans & Policy Integration)

3/19/93
Date

Margaret J. James
BILLY B. WILLIAMS
Executive Director
(Procurement)

3/24/93
Date

APPENDIX B
LIST OF DATA ELEMENTS ORDERED BY SOURCE

Appendix B - List of Data Elements Ordered By Source

listing notes:

* denotes data element which has previously been identified as new and already planned for incorporation into existing database

** denotes new data element requirement

DGSC.P.DVRS0A1	*Type*	*Width*	*Decimals*	*Reports*
cage	Character	5		5
contract_delv_date	Numeric	6		5
contract_mod_number	Character	6		5
correction_date	Numeric	6		5
nsn	Character	15		5
project_action_code	Character	2		5
qty_due	Numeric	9		5
qty_recpt	Numeric	9		5
qty_ship	Numeric	9		5
reason_delay_code	Character	2		5
receipt_date	Numeric	6		5
ship_date	Numeric	6		5
term_code	Character	1		5
variance_code	Character	3		5

DGSC.P.DVRS101	*Type*	*Width*	*Decimals*	*Reports*
cage	Character	5		4,7
cont_line**	Character	23		7
del_date	Numeric	5		4,7
fsc	Character	4		4,7
ship_date	Numeric	5		4,7

DGSC.P.DVRS701	*Type*	*Width*	*Decimals*	*Reports*
cage	Character	5		4,7
piin_clin	Character	23		4,7
caus_cd	Character	2		4,7
clos_dt	Numeric	5		4,7
test_dtd	Numeric	5		4,7
disc_cd	Character	2		7
doc_typ	Character	1		7
fsc	Character	4		4,7
no_crit	Numeric	6		4,7
no_maj	Numeric	6		4,7
no_min	Numeric	6		4,7
no_tested	Numeric	6		4,7

dpacs-arch.bidven	*Type*	*Width*	*Decimals*	*Reports*
award_date	Date			8,9
award_total*	Numeric	16	2	1,3,6
business_size	Character	1		3,4,8,9
business_type	Character	2		3,8,9
buy_number	Character	13		1,4,6,8,9
cage_code	Character	5		1,2,3,4,6, 8,9
delivery_score**	Numeric	5	1	3
dlvy_weight*	Numeric	3		9
dvrs_rating	Numeric	5	1	4,6,8,9
extend_bid_price**	Numeric	16	2	1,3,6,8
qual_weight*	Numeric	3		9
quality_score**	Numeric	5	1	3

dpacs-arch.mudkey	*Type*	*Width*	*Decimals*	*Reports*
buy_number	Character	13		8,9
set_aside_flag	Character	1		8,9

dpacs-dvrs.vr_crate	*Type*	*Width*	*Decimals*	*Reports*
avg_days_late	Numeric	5	1	2
cage_code	Character	5		2,3
contract_lines	Numeric	6		2
delivery_score	Numeric	5	1	2,3
labtest_deficits	Numeric	5	1	2
on_time_pct	Numeric	5	1	2
overall_score	Numeric	5	1	2,3
package_deficits	Numeric	5	1	2
product_deficits	Numeric	5	1	2
quality_score	Numeric	5	1	2,3
score_date**	Date			2

dpacs-dvrs.vrf_rate	*Type*	*Width*	*Decimals*	*Reports*
avg_days_late	Numeric	5	1	2
delivery_score	Numeric	5	1	2
fsc_code	Character	4		2,10
labtest_deficits	Numeric	5	1	2
on_time_pct	Numeric	5	1	2
overall_score	Numeric	5	1	2,4,10
package_deficits	Numeric	5	1	2
product_deficits	Numeric	5	1	2
quality_score	Numeric	5	1	2
score_date**	Date			2,4,10

dpacs-dvrs.vrfcrate	*Type*	*Width*	*Decimals*	*Reports*
avg_days_late	Numeric	5	1	2
cage_code	Character	5		2,3
critical_quality_deficiencies**	Numeric	6		3
delivery_score	Numeric	5	1	2,3
fsc_code	Character	4		2,3
labtest_deficits	Numeric	5	1	2
on_time_pct	Numeric	5	1	2
overall_score	Numeric	5	1	2,3
package_deficits	Numeric	5	1	2
pct_on_time_clins**	Numeric	5	1	3
product_deficits	Numeric	5	1	2
quality_score	Numeric	5	1	2,3
score_date**	Date			2

dpacs.dwarfdb	*Type*	*Width*	*Decimals*	*Reports*
abvm_clause*	Character	1		1,2,3,6,7,8,9,11
abvm_dlvy_qual*	Character	1		9
abvm_pric_perf*	Character	1		9
abvm_qual_dlvy*	Character	1		9
buy_number	Character	13		1,4,6,8,9,11
cage_code	Character	5		1,2,3,4,6,8,9,11
contract_no	Character	13		7,11
differential**	Numeric	16	2	1,3,7,8,11
nsn	Character	15		1,2,3,4,6,11
total_cost_price	Numeric	16	2	1,3,6,8,9

dpacs.opencon	*Type*	*Width*	*Decimals*	*Reports*
cage_code	Character	5		11
clin	Character	6		11
contract_no	Character	17		11
delivery_date	Date			11
nsn	Character	15		11

dpacs.soldata	*Type*	*Width*	*Decimals*	*Reports*
abvm_clause**	Character	1		1,2,3,6
buy_no	Character	13		1,2,3,6

dpacs.vendor	*Type*	*Width*	*Decimals*	*Reports*
cage_code	Character	5		3,11
manufacturer** (type_enterprise)	Character	1		3
mfg_floor_space	Numeric	10		3
quality_vendor_pgm**	Character	1		3

type_organization	Character	1		3
type_owner	Character	1		3
vend_name	Character	30		3,11

dpacs.workperf	*Type*	*Width*	*Decimals*	*Reports*
avg_palt	Numeric	3		4
week_number	Numeric	3		4

GOR.X.DVRS00CD	*Type*	*Width*	*Decimals*	*Reports*
challenge_compl_date	Numeric	6		5
challenge_decsn_date	Numeric	6		5
challenge_recv_date	Numeric	6		5
denial_code	Character	1		5
fsc	Character	4		5

GOR.X.DVRS00CQ	*Type*	*Width*	*Decimals*	*Reports*
challenge_compl_date	Numeric	6		5
challenge_decsn_date	Numeric	6		5
challenge_recv_date	Numeric	6		5
denial_code	Character	1		5
fsc	Character	4		5

GOR.X.DVRS00DD	*Type*	*Width*	*Decimals*	*Reports*
challenge_compl_date	Numeric	6		5
challenge_decsn_date	Numeric	6		5
challenge_recv_date	Numeric	6		5
denial_code	Character	1		5
fsc	Character	4		5

GOR.X.DVRS00PD	*Type*	*Width*	*Decimals*	*Reports*
challenge_compl_date	Numeric	6		5
challenge_decsn_date	Numeric	6		5
challenge_recv_date	Numeric	6		5
denial_code	Character	1		5
fsc	Character	4		5

GOR.X.DVRS00PQ	*Type*	*Width*	*Decimals*	*Reports*
challenge_compl_date	Numeric	6		5
challenge_decsn_date	Numeric	6		5
challenge_recv_date	Numeric	6		5
denial_code	Character	1		5
fsc	Character	4		5

GOR.X.DVRS00UD	*Type*	*Width*	*Decimals*	*Reports*
challenge_compl_date	Numeric	6		5
challenge_decsn_date	Numeric	6		5
challenge_recv_date	Numeric	6		5

denial_code	Character	1		5
fsc	Character	4		5

GOR.X.DVRS00UQ	*Type*	*Width*	*Decimals*	*Reports*
challenge_compl_date	Numeric	6		5
challenge_decsn_date	Numeric	6		5
challenge_recv_date	Numeric	6		5
denial_code	Character	1		5
fsc	Character	4		5

APPENDIX C
NEW DATA ELEMENT REQUIREMENTS

Appendix C - New Data Element Requirements

Filename/Data Element	*Type*	*Width*	*Decimals*	*Reports*
DGSC.P.DVRS101 cont_line**	Character	23		7
dpacs-arch.bidven delivery_score**	Numeric	5	1	3
dpacs-arch.bidven extend_bid_price**	Numeric	16	2	1,3,6,8
dpacs-arch.bidven quality_score**	Numeric	5	1	3
dpacs-dvrs.vr_crate score_date**	Date			2
dpacs-dvrs.vrf_rate score_date**	Date			2,4
dpacs-dvrs.vrfcrate critical_quality_deficiencies**	Numeric	6		3
dpacs-dvrs.vrfcrate pct_on_time_clins**	Numeric	5	1	3
dpacs-dvrs.vrfcrate score_date**	Date			2
dpacs.dwarfdb differential**	Numeric	16	2	1,3,7,8,11
dpacs.soldata abvm_clause**	Character	1		1,2,3,6
dpacs.vendor manufacturer** (type_enterprise)	Character	1		3
dpacs.vendor quality_vendor_pgm**	Character	1		3

New data element requirements (by report)

1. Summary ABVM Application Statistics

dpacs.soldata.abvm_clause
dpacs-arch.bidven.extend_bid_price
dpacs.dwarfdb.differential

2. Monthly ABVM Component Score Statistics

dpacs.soldata.abvm_clause
dpacs-dvrs.vrfcrate.score_date
dpacs-dvrs.vr_crate.score_date
dpacs-dvrs.vrf_rate.score_date

3. Quality Vendor Program Information

dpacs.soldata.abvm_clause
dpacs-arch.bidven.extend_bid_price
dpacs-arch.bidven.delivery_score
dpacs-arch.bidven.quality_score
dpacs.dwarfdb.differential
dpacs.vendor.quality_vendor_pgm
dpacs.vendor.manufacturer **(type_enterprise)**
dpacs-dvrs.vrfcrate.critical_quality_deficiencies
cpacs-dvrs.vrfcrate.pct_on_time_clins

4. Center ABVM Implementation "Benefits" Indicators

dpacs-dvrs.vrf_rate.score_date

5. Challenge Statistics - none

6. ABVM Buyer Performance

dpacs.soldata.abvm_clause
dpacs-arch.bidven.extend_bid_price

7. ABVM Program Award Performance

dpacs.dwarfdb.differential
DGSC.P.DVRS101 -- cont_line (23)

8. Center ABVM Statistics

dpacs-arch.bidven.extend_bid_price
dpacs.dwarfdb.differential

9. Center ABVM Statistics by Solicitation Consideration Factors - none

10. ABVM Score Trends

dpacs-dvrs.vrf_rate.score_date

11. Near-term Deliveries - ABVM Differential Awards

dpacs.dwarfdb.differential

APPENDIX D
CURRENT DATA ELEMENTS REQUIRING ARCHIVAL

Appendix D - Current Data Elements Requiring Archival

DGSC.P.DVRS101
cage
cont_line**
del_date
fsc
ship_date

DGSC.P.DVRS701
cage
piin_clin
caus_cd
clos_dt
test_dtd
disc_cd
doc_typ
fsc
no_crit
no_maj
no_min
no_tested

dpacs.dwarfdb
abvm_clause*
abvm_dlvy_qual*
abvm_pric_perf*
abvm_qual_dlvy*
buy_number
cage_code
contract_no
differential**
nsn
total_cost_price

dpacs.soldata
abvm_clause**
buy_no

dpacs-dvrs.vrf_rate
fsc_code
overall_score
score_date**

APPENDIX E
STANDARD REPORT FORMATS

Acronyms/Abbreviations Used in Report Formats

ABVM	Automated Best Value Model
CAGE	Contractor and Government Entity
CDD	Contract Delivery Date
CLIN	Contract Line Item Number
DSC	DLA Supply Center
FSC	Federal Supply Class
LDV	Low Dollar Value
NSN	National Stock Number
SDB	Small, Disadvantaged Business

1. Summary ABVM Application Statistics

Federal Supply Class	Number of Solicitations	Percent ABVM Clause	Number of Awards	Percent with Differential	Average Differential ($)	Average Differential (%)
1000	342	95.4	375	15.2	$512	24.30%
1510	960	10.3	1088	12.5	$32	4.50%
DSC Total	1302	32.7	1463	13.2	$400	19.69%

Sort = FSC

2. Monthly ABVM Component Score Statistics

FSC = XXXX	ABVM score	Non-ABVM score	Total score
On-time delivery			
* awarded average	95	81	88
* bid average	83	82	83
* FSC average			85
* DSC average			78
Days late			
* awarded average	97	95	96
* bid average	94	92	93
* FSC average			90
* DSC average			82
Product deficiencies			
* awarded average	89	75	82
* bid average	73	70	72
* FSC average			70
* DSC average			78
Packaging deficiencies			
* awarded average	88	89	89
* bid average	87	92	90
* FSC average			89
* DSC average			78
Lab test results			
* awarded average	94	65	77
* bid average	75	70	73
* FSC average			66
* DSC average			82

	ABVM score	Non-ABVM score	Total score
DELIVERY			
* awarded average	96	88	92
* bid average	88	87	88
* FSC average			88
* DSC average			80
QUALITY			
* awarded average	90	76	83
* bid average	78	74	78
* FSC average			75
* DSC average			79
OVERALL			
* awarded average	92	83	87
* bid average	85	82	83
* FSC average			81
* DSC average			79

Sort = FSC

3. Quality Vendor Program Information

CAGE	Vendor Name	FSC	No. of Contracts	No. CLINs awarded	Dollar value of awards	Business size	Socio-economic status	No. differential awards	ABVM Overall Score	ABVM Quality Score	Critical quality deficiencies	ABVM Delivery Score	% CLINs on-time	Manuf?
ABCDE	Fiberbrain Corp.	5290	29	53	$30,157	small		5	98.2	100.0	N	96.4	87.3	N
FGHIJ	Mega-industries	all	706	1022	$572,091	large		34	99.1	98.6		100.0	100.0	
		5290	415	672	$457,091			14	99.3	98.6	N	100.0	100.0	Y
		6090	291	350	$115,000			20	98.7	98.5	N	100.0	100.0	N

Sort = CAGE, FSC

4. Center ABVM Implementation "Benefits" Indicators

FSC = XXXX

	Pre-ABVM	Post-ABVM	Percent change	
Delinquency Rate	27.3	24.7	-9.52%	*used in Economic Analysis*
Product Nonconformance Rate	6.3	5.3	-15.87%	*used in Economic Analysis*
Packaging Nonconform Rate	15.3	13.1	-14.38%	*used in Economic Analysis*
% Terminations (vendor)	1.2	1.1	-8.33%	
Average Days Delinquent	46.1	44.3	-3.90%	
ABVM Scores	86.3	87.1	0.93%	
No. of participating vendors	527	460	-12.71%	*unique participating vendors*
% Awards to "New" Vendors	12.8	12	-6.25%	
"new" small	1.5	0.8	-46.67%	
"new" large	5.3	6.2	16.98%	
"new" SDB	3.9	2.8	-28.21%	
"new" woman-owned	2.1	2.2	4.76%	
Procurement Administrative Lead Time	43.2	47.5	9.95%	
% Awards to Small Business	13.5	12.9	-4.44%	

Sort = FSC

5. Challenge Statistics

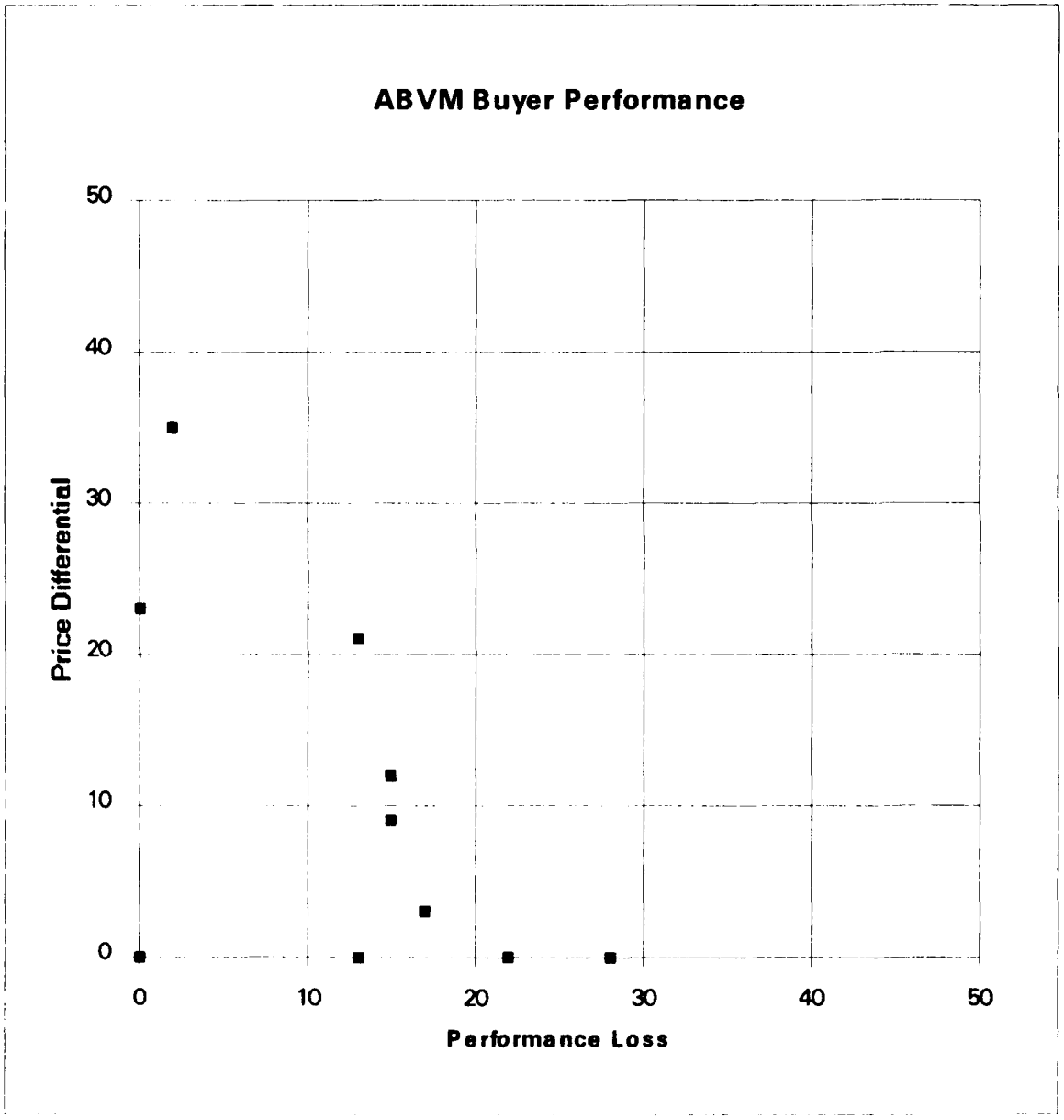
FSC = XXXX

	Quality	Delivery
# Challenges:		
Received		
Approved		
Denied		
Pending		
Database Corrections		
Average # Days Required:		
Approval		
Denial		
Database Correction		
Type of Change:	Modify	
Contract Mod Number		
CAGE		
NSN		
Contract Delivery Date		
Ship Date		
Receipt Date		
Variance Code		
Reason for Delay Code		
Termination Code		
Project Action Code		
Quantity Due		
Quantity Received		
Quantity Shipped		

Sort = FSC

6. ABVM Buyer Performance

FSC = XXXX



Sort = FSC

7. ABVM Program Award Performance

FSC = XXXX	ABVM Differential	ABVM No Differential	Non-ABVM
Delinquency Rate	2.1	3.5	5.1
Average Days Delinquent	11.2	26.9	46.3
Product Deficiency Rate	2.2	4.1	4.5
Packaging Deficiency Rate	1.2	8.3	8.5
Lab Test Deficiency Rate	9.8	16.1	15.6

Sort = FSC

8. Center ABVM Statistics

FSC = XXXX	*Awards*		*Dollar Value of Awds*		*Ave. Difference Between Low Offer/Awardee*	
	Total No.	% LDV	Total $	% LDV	$	% ($)
Small Business Awards						
Low offer						
Low offer/highest score						
Low offer/not highest score						
Other than low offer						
New offer/no score						
New offer/not lowest price						
Small Business Set-Asides						
Low offer						
Low offer/highest score						
Low offer/not highest score						
Other than low offer						
New offer/no score						
New offer/not lowest price						
Small/SDB Set-Asides						
Low offer						
Low offer/highest score						
Low offer/not highest score						
Other than low offer						
New offer/no score						
New offer/not lowest price						
Large Business						
Low offer						
Low offer/highest score						
Low offer/not highest score						
Other than low offer						
New offer/no score						
New offer/not lowest price						

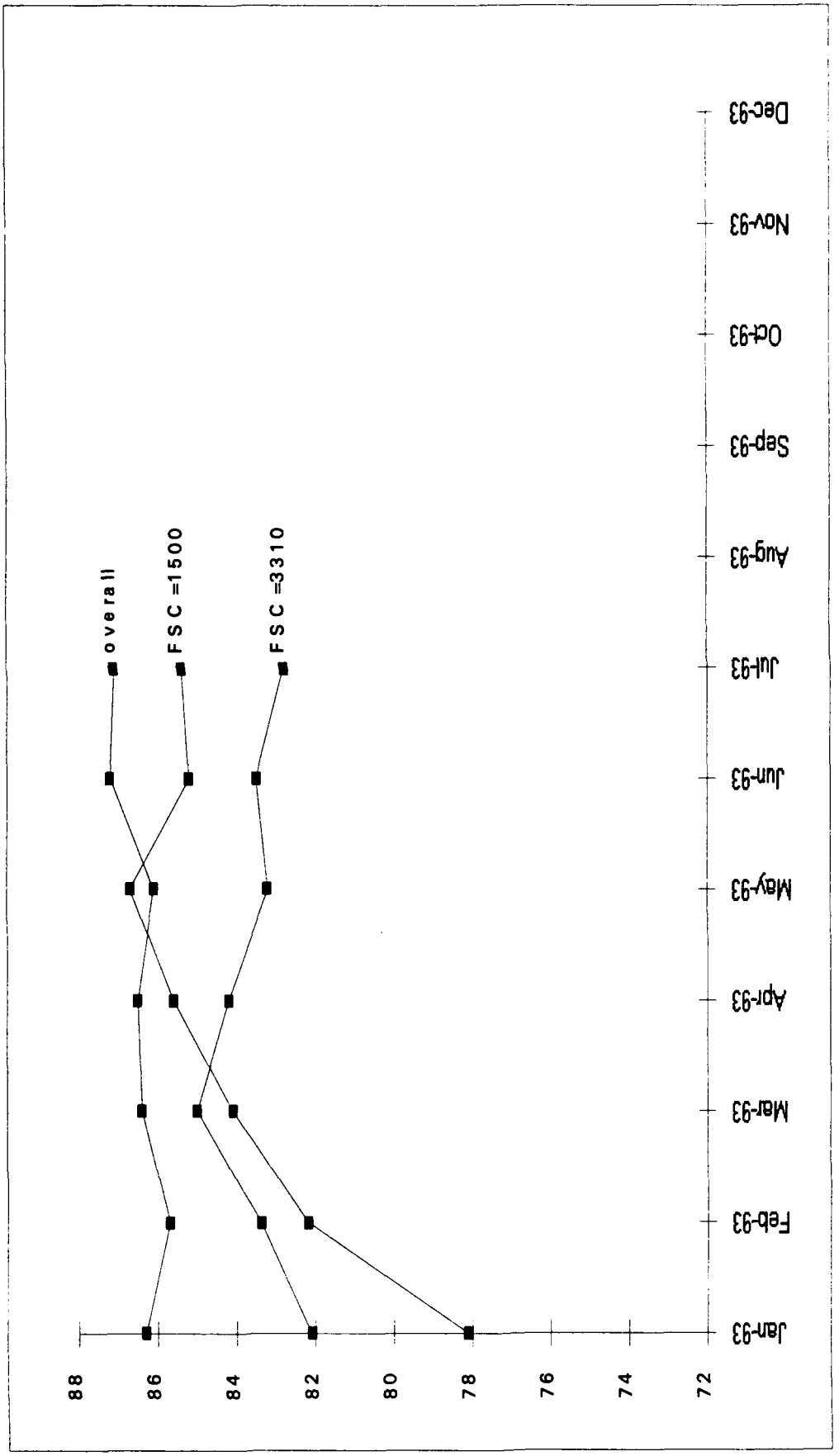
Sort = FSC

9. Center ABVM Statistics by Solicitation Consideration Factors

FSC = XXXX	Total Proc Actions		Small Business		Small Set-Aside		Small/SDB Set-Aside		Large Business	
	#	$	#	$	#	$	#	$	#	$
Price Only (Non-ABVM)										
Price = Performance (ABVM)										
delivery = quality										
delivery > quality										
delivery < quality										
Price > Performance (ABVM)										
delivery = quality										
delivery > quality										
delivery < quality										
Price < Performance (ABVM)										
delivery = quality										
delivery > quality										
delivery < quality										

Sort = FSC

10. ABVM Score Trends



11. Near-term Deliveries - ABVM Differential Awards

Date: 1 Feb 93
FSC: XXXX

CDD	CAGE	Contractor Name	Contract #/CLIN	NSN
XX XX XX	XXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXX / XXXX	XXXX-XX-XXX-XXXX

Sort = CAGE, CDD, Contract #, CLIN

APPENDIX F
PSEUDOCODE COMPUTATIONS TO PRODUCE REPORTS

Standard Report Computations -

This appendix describes the computations required to produce the standard reports described in this functional description. The computations are described in a form of pseudocode. The data elements are generally described by the full data element name in the format *system.table.element*. Local variables used to store results of computations are generally of the simple form *variable*, except in those cases where additional hierarchy or structure is required to more easily describe the results.

1. Summary ABVM Application Statistics

Computations:

* first, compute differential for ABVM awards

```
for each award where dpacs.dwarfdb.abvm_clause='y':
   low_bid = min(dpacs-arch.bidven.extend_bid_price)
   differential = dpacs.dwarfdb.total_cost_price - low_bid
   differential_percent = differential  /  dpacs.dwarfdb.total_cost_price

for each FSC,
   sum number_solicitations
   sum number_abvm_solicitations
   sum number_awards
   percent_abvm = number_abvm_solicitations / number_solicitations
   sum number_differential (awards where differential > 0)
   percent_differential_awards = number_differential / number_awards
   total_differential = sum(differential)
   total_differential_percent = sum(differential_percent)
   average_differential = total_differential / number_differential
   average_differential_percent = total_differential_percent / number_differential

for all FSC's,
   sum/average over all FSC's
```

2. Monthly ABVM Component Score Statistics

Computations:

```
by FSC
for each buy,   identify type (whether an ABVM solicitation)

for each type of solicitation (ABVM or non-ABVM) & total
   identify bidding vendors (dpacs-arch.bidven.cage_code)
   sum number_vendors_bidding
   bid_score.[component] = sum(dpacs-dvrs.vrfcrate.[component]) / number_vendors_bidding

(where [component] is on_time_pct, avg_days_late, product_deficits, package_deficits, labtest_deficits, delivery_score, quality_score,
overall_score)

      identify winning vendor (dpacs.dwarfdb.cage_code)

for each FSC
   sum number_solicitations
   sum number_abvm_solicitations
   fsc.bid_score.[component] = sum(bid_score.[component]) / number_solicitations
   fsc-abvm.bid_score.[component] = appropriate sum(bid_score.[component]) / number_abvm_solicitations
   fsc-non.bid_score.[component] = appropriate sum(bid_score.[component]) / (number_solicitations - number_abvm_solicitations)
   fsc.award_score.[component] = sum(dpacs-dvrs.vrfcrate.[component] -> dpacs.dwarfdb.cage_code) / number_solicitations
   fsc-abvm.award_score.[component] = appropriate sum . . . .
   fsc-non.award_score.[component] = appropriate sum . . . .
   fsc.population.[component] = dpacs-dvrs.vrf_rate.[component]
   center.population.[component] = sum(dpacs-dvrs.vr_crate.[component] -> for each CAGE) / number_cages
```

3. Quality Vendor Program Information

Computations:

```
for each award where dpacs.dwarfdb.abvm_clause='y':
   low_bid = min(dpacs-arch.bidven.extend_bid_price)
   differential = dpacs.dwarfdb.total_cost_price - low_bid

for each CAGE,   sum number_awards (where dpacs.dwarfdb.cage_code = CAGE)
   sum number_differential (awards where differential > 0)
   award_value = sum(dpacs.dwarfdb.total_cost_price)
```

4. Center ABVM Implementation "Benefits" Indicators

Computations:

```
for each FSC and time period (pre- and post-ABVM)   ???
   days_late = dvrs101.ship_date - dvrs101.del_date
   if days_late > 0 then late=.true.
   if days_late < 0 then set days_late=0
   sum number_awards
   sum number_late_awards (where days_late > 0)
   total_days_late = sum(days_late)
   delinq_rate = number_late_awards / number_awards
   delinq_days = total_days_late / number_late_awards
   lab_defic = no_crit + no_maj + no_min
   total_tested = sum(no_tested)
   total_lab_defic = sum(lab_defic)
   lab_rate = total_lab_defic / total_tested
   sum number_prod_defic (where test_dtd='     ' and not a packaging deficiency (see below))
   sum number_pkg_defic (where test_dtd='     ' and doc_typ=(2 or 4 thru 9) and disc_cd=(P0 thru P7 or T4 or T6))
   prod_rate = number_prod_defic / number_awards
   pkg_rate = number_pkg_defic / number_awards
   number_vendors = unique sum (dpacs-arch.bidven.cage_code)
   number_small_awds = sum awards where (dpacs.dwarfdb.cage_code = dpacs-arch.bidven.cage_code -> dpacs-arch.bidven.business_size=small)
   pct_small_awds = number_small_awds / number_awards
   number_new_awds = sum awards where (dpacs.dwarfdb.cage_code = dpacs-arch.bidven.cage_code -> dpacs-arch.bidven.dvrs_rating=0.0)
   pct_new_awds = number_new_awds / number_awards
```

Notes:

* This report should be generated by FSC.
* Must ensure that only FSC's with "clean" data are examined, to ensure that any changes are not attributable merely to the sanitization of data.
* Must still identify criteria to be used to identify "regular" vendors.
* Might use lack of an ABVM score to identify "new" vendors.

5. Challenge Statistics

Computations:

```
by FSC:
   number_qual_recv = sum of records in quality files (where challenge_recv_date > rpt_start_date)
   number_qual_appr = sum of records in dvrs00pq & dvrs00cq (where challenge_decsn_date > rpt_start_date and denial_code -= 'y')
   number_qual_denied = sum of records in dvrs00pq & dvrs00cq (where challenge_decsn_date > rpt_start_date and denial_code = 'y')
   number_qual_pend = sum of records in dvrs00pq (where challenge_decsn_date > rpt_start_date)
   number_qual_corr = sum of records in dvrs00cq (where challenge_compl_date > rpt_start_date)
   number_delv_recv = sum of records in delivery files (where challenge_recv_date > rpt_start_date)
   number_delv_appr = sum of records in dvrs00pd & dvrs00cd (where challenge_decsn_date > rpt_start_date and denial_code -= 'y')
   number_delv_denied = sum of records in dvrs00dd (where challenge_decsn_date > rpt_start_date and denial_code = 'y')
   number_delv_pend = sum of records in dvrs00pd (where challenge_decsn_date > rpt_start_date)
   number_delv_corr = sum of records in dvrs00cd (where challenge_compl_date > rpt_start_date)

for denial_code = 'y'
   delv_denial_days = dvrs00dd.challenge_decsn_date - dvrs00dd.challenge_recv_date (where challenge_decsn_date > rpt_start_date)

for denial_code <> 'y'
   delv_approval_days = (dvrs00pd or dvrs00cd).challenge_decsn_date - (dvrs00pd or dvrs00cd).challenge_recv_date (where challenge_decsn_date >
rpt_start_date)
   qual_approval_days = (dvrs00pq or dvrs00cq).challenge_decsn_date - (dvrs00pq or dvrs00cq).challenge_recv_date (where challenge_decsn_date >
rpt_start_date)
   delv_correction_days = dvrs00cd.challenge_compl_date - dvrs00cd.challenge_decsn_date (where challenge_compl_date > rpt_start_date)
   qual_correction_days = dvrs00cq.challenge_compl_date - dvrs00cq.challenge_decsn_date (where challenge_compl_date > rpt_start_date)

average_delv_denial_days = sum(delv_denial_days) / number_delv_denied
average_delv_approval_days = sum(delv_approval_days) / number_delv_appr
average_delv_correction_days = sum(delv_correction_days) / number_delv_corr
average_qual_approval_days = sum(qual_approval_days) / number_qual_appr
average_qual_correction_days = sum(qual_correction_days) / number_qual_corr

in the file dvrs0a1, count occurrences of non-blank fields
   number_cmod = sum(non-blank contract_mod_number)
   number_cage = sum(non-blank cage)
   number_nsn = sum (non-blank nsn)
   number_cdd = sum(non-blank contract_delv_date)
   number_ship = sum(non-blank ship_date)
   number_recv = sum(non-blank receipt_date)
   number_var = sum(non-blank variance_code)
   number_rsn = sum(non-blank reason_delay_code)
   number_term = sum(non-blank term_code)
   number_proj = sum(non-blank project_action_code)
   number_q_due = sum(non-blank qty_due)
   number_q_rcv = sum(non-blank qty_recpt)
   number_q_shp = sum(non-blank qty_ship)
```

6. ABVM Buyer Performance

Computations:

```
for each FSC
for each award with dpacs.dwarfdb.abvm_clause = 'y'
   low_bid = min(dpacs-arch.bidven.extend_bid_price)
   differential = dpacs.dwarfdb.total_cost_price - low_bid
   differential_percent = differential  /  low_bid
   hi_perf = max(dpacs-arch.bidven.drvs_rating)
   perf_loss = hi_perf - dpacs-arch.bidven.drvs_rating(where dpacs-arch.bidven.cage_code = dpacs.dwarfdb.cage_code)
   perf_loss_percent = perf_loss / hi_perf

plot differential_percent versus perf_loss_percent
```

7. ABVM Program Award Performance

Computations:

```
for each FSC
   three categories identified by
   ABVM differential:   dpacs.dwarfdb.abvm_clause = 'y' and dpacs.dwarfdb.differential > 0
   ABVM no-differential:   dpacs.dwarfdb.abvm_clause = 'y' and dpacs.dwarfdb.differential = 0
   Non-ABVM:      dpacs.dwarfdb.abvm_clause = 'n'

match dvrs101.cont_line to dpacs.dwarfdb.contract_no
for each category:
   days_late = dvrs101.ship_date - dvrs101.del_date
   if days_late > 0 then late=.true.
   if days_late < 0 then set days_late=0
   sum number_awards
   sum number_late_awards (where days_late > 0)
   total_days_late = sum(days_late)
   delinq_rate = number_late_awards / number_awards
   delinq_days = total_days_late / number_late_awards
match dvrs701.piin_clin to dpacs.dwarfdb.contract_no

for each category:
   lab_defic = no_crit + no_maj + no_min
   total_tested = sum(no_tested)
   total_lab_defic = sum(lab_defic)
   lab_rate = total_lab_defic / total_tested
   sum number_prod_defic (where test_dtd='     ' and not a packaging deficiency (see below))
   sum number_pkg_defic (where test_dtd='     ' and doc_typ=(2 or 4 thru 9) and disc_cd=(P0 thru P7 or T4 or T6))
   sum number_awards
   prod_rate = number_prod_defic / number_awards
   pkg_rate = number_pkg_defic / number_awards
```

Notes:

* Must match ABVM award information to performance information.

8. Center ABVM Statistics

Computations:

```
for start_date <= dpacs-arch.bidven.award_date <= end_date
for each socio-economic category      ?? how to determine each category ??
{
for each award where dpacs.dwarfdb.abvm_clause='y':
   low_bid = min(dpacs-arch.bidven.extend_bid_price)
   differential = dpacs.dwarfdb.total_cost_price - low_bid
   differential_percent = differential  /  dpacs.dwarfdb.total_cost_price
   score_delta = dpacs-arch.bidven.dvrs_rating(dpacs.dwarfdb.cage_code) - dpacs-arch.bidven.dvrs_rating(low_bid(dpacs-arch.bidven.cage_code))

sum number_awards
sum number_low_dollar_value (where dpacs.dwarfdb.total_cost_price < LDV threshhold)
percent_number_low_dollar = number_low_dollar_value / number_awards
total_award_dollars = sum(dpacs.dwarfdb.total_cost_price)
value_low_dollar_value = sum(dpacs.dwarfdb.total_cost_price < LDV threshhold)
percent_value_low_dollar = value_low_dollar_value / total_award_dollars
total_differential = sum(differential)
ave_diff = total_differential / number_awards
ave_diff_pct = total_differential / total_award_dollars
total_score_delta = sum(score_delta)
ave_score_delta = total_score_delta / number_awards
}
```

9. Center ABVM Statistics by Solicitation Consideration Factors

```
Computations:

establish reporting period (start_date, end_date)
for start_date <= dpacs-arch.bidven.award_date <= end_date

determine consideration factor category for each award:

if (dpacs.dwarfdb.abvm_clause='n') then
   price_only = .true.
else if (dpacs.dwarfdb.abvm_clause='y' and dpacs.dwarfdb.abvm_pric_perf = 'X')    then
   price_equl_perf = .true.
else if (dpacs.dwarfdb.abvm_clause='y' and dpacs.dwarfdb.abvm_pric_perf='Y') then
   price_grtr_perf = .true.
else if (dpacs.dwarfdb.abvm_clause='y' and dpacs.dwarfdb.abvm_pric_perf='Z') then
   begin
      price_less_perf = .true.

      if (dpacs.bidven.dlvy_weight = dpacs.bidven.qual_weight) then
         delv_equl_qual = .true.
      else if (dpacs.bidven.dlvy_weight > dpacs.bidven.qual_weight) then
         delv_grtr_qual = .true.
      else if (dpacs.bidven.dlvy_weight < dpacs.bidven.qual_weight) then
         delv_less_qual = .true.
      endif
   end
endif

for each consideration factor category
{
for each socio-economic category      ?? how to determine each category ??
   sum number_awards
   total_award_dollars = sum(dpacs.dwarfdb.total_cost_price)
}
```

10. ABVM Score Trends

```
Computations:

Plot dpacs-dvrs.vrf_rate.overall_score versus time (dpacs-dvrs.vrf_rate.score_date).

Notes:
*  Must determine how to aggregate scores across FSC's.  May require additional computation of raw data.
```

11. Near-term Deliveries - ABVM Differential Awards

```
Computations:

for each FSC,
   if (dpacs.dwarfdb.abvm_clause = 'y') then
      if (dpacs.dwarfdb.differential > 0.0) and (dpacs.dwarfdb.cage_code = dpacs.opencon.cage_code) then
        provide listing sorted by
           dpacs.opencon.cage_code
           dpacs.opencon.delivery_date
           dpacs.opencon.contract_no
           dpacs.opencon.clin
      endif
   endif
```

APPENDIX G
PARTIAL PROTOTYPE OF ABVM DSS IMPLEMENTED IN DBASE IV

1 - Data Dictionary

2 - Program Source Code Listing

3 - Sample Screens

System: ABVM Decision Support System
Author:
08/12/93 15:17:43
Database Structure Summary

---

21 databases in the system
   DWARFDB.DBF
   BIDVEN.DBF
   VRFCRATE.DBF
   AWDINFO.DBF
   VR-CRATE.DBF
   VRF-RATE.DBF
   VENDOR.DBF
   SOLDATA.DBF
   WORKPERF.DBF
   DVRS101.DBF
   DVRS701.DBF
   DVRS00UD.DBF
   DVRS00UQ.DBF
   DVRS00PD.DBF
   DVRS00PQ.DBF
   DVRS00DD.DBF
   DVRS00CQ.DBF
   DVRS00CD.DBF
   DVRS0A1.DBF
   MUDKEY.DBF
   OPENCON.DBF

Structure for database : DWARFDB.DBF
Number of data records :   5
        Last updated : 08/12/93 at 14:36

Field	Field name	Type	Width	Dec	Start	End
1	ABVM_CLAUS	Character	1		1	1
2	ABVM_D_Q	Character	1		2	2
3	ABVM_P_P	Character	1		3	3
4	ABVM_Q_D	Character	1		4	4
5	BUY_NO	Character	13		5	17
6	CAGE_CODE	Character	5		18	22
7	CONTRACT_N	Character	13		23	35
8	DIFFERENTL	Numeric	16	2	36	51
9	NSN	Character	15		52	66
10	TOTAL_COST	Numeric	16	2	67	82
** Total **			83			

This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\AWDBUY.NDX (buy_no)
       : C:\DATA\DBASE\AWDNSN.NDX (nsn)
       : C:\DATA\DBASE\AWDCAGE.NDX (cage_code+substr(nsn,1,4))
       : C:\DATA\DBASE\AWDCONT.NDX (substr(nsn,1,4)+contract_n)

Used by: RPT-1.PRG
       : RPT-2.PRG
       : RPT-4.PRG
       : RPT-6.PRG
       : RPT-8.PRG
       : RPT-9.PRG
       : RPT-11.PRG
       : FSC_STATS   (procedure in C:\DATA\DBASE\RPT-1.PRG)
       : SETUP_FILS   (procedure in C:\DATA\DBASE\RPT-3.PRG)
       : SETP_FILS   (procedure in C:\DATA\DBASE\RPT-7.PRG)

```
Structure for database : BIDVEN.DBF
Number of data records :        15
        Last updated : 07/28/93 at 10:04
Field  Field name  Type        Width    Dec    Start      End
    1  AWARD_DT    Date            8              1        8
    2  AWARD_TOTL  Numeric        16      2       9       24
    3  BUS_SIZE    Character       1              25       25
    4  BUS_TYPE    Character       2              26       27
    5  BUY_NO      Character      13              28       40
    6  CAGE_CODE   Character       5              41       45
    7  DELIV_SCOR  Numeric         5      1       46       50
    8  DLVY_WGT    Numeric         3              51       53
    9  DVRS_RATIN  Numeric         5      1       54       58
   10  EXTEND_BID  Numeric        16      2       59       74
   11  QUAL_WGT    Numeric         3              75       77
   12  QUAL_SCOR   Numeric         5      1       78       82
** Total **                      83


This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\BIDBUY.NDX  (buy_no)

Used by: RPT-1.PRG
       : RPT-4.PRG
       : RPT-6.PRG
       : RPT-8.PRG
       : RPT-9.PRG
       : COMP_BIDS                (procedure in C:\DATA\DBASE\RPT-2.PRG)
       : OPEN_OTHR                (procedure in C:\DATA\DBASE\RPT-2.PRG)
       : GET_OTHR                 (procedure in C:\DATA\DBASE\RPT-3.PRG)

--------------------------------------------------------------------------------


Structure for database : VRFCRATE.DBF
Number of data records :        11
        Last updated : 08/12/93 at  7:14
Field  Field name  Type        Width    Dec    Start      End
    1  AVG_DAYS_L  Numeric         5      1       1        5
    2  CAGE_CODE   Character       5              6       10
    3  CRIT_QUAL   Numeric         6             11       16
    4  DELIV_SCOR  Numeric         5      1       17       21
    5  FSC_CODE    Character       4              22       25
    6  LABTEST_DF  Numeric         5      1       26       30
    7  ON_TIME_PC  Numeric         5      1       31       35
    8  OVERALL_SC  Numeric         5      1       36       40
    9  PACKAGE_DF  Numeric         5      1       41       45
   10  PCT_CLINS   Numeric         5      1       46       50
   11  PRODUCT_DF  Numeric         5      1       51       55
   12  QUA' SCOR   Numeric         5      1       56       60
   13  SCORE_DT    Date            8              61       68
** Total **                      69


This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\VRFCAGE.NDX  (cage_code+fsc_code)

Used by: RPT-2.PRG
       : SETUP_FILS               (procedure in C:\DATA\DBASE\RPT-3.PRG)

--------------------------------------------------------------------------------
```

```
Structure for database : AWDINFO.DBF
Number of data records :          5
          Last updated : 08/12/93 at 14:37
Field  Field name  Type        Width    Dec    Start      End
    1  BUY_NO      Character      13             1       13
    2  FSC         Character       4            14       17
    3  TOT_BIDS    Numeric         5            18       22
    4  RTD_BIDS    Numeric         5            23       27
    5  BID_ABVM    Numeric         5      1     28       32
    6  ABVM_CLAUS  Character       1            33       33
    7  DIFFERENTL  Numeric        16      2     34       49
    8  TOTAL_COST  Numeric        16      2     50       65
** Total **                      66


This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\AWDKEY.NDX   (buy_no)
       : C:\DATA\DBASE\AWDCAGE.NDX  (cage_code+substr(nsn,1,4))

Used by: RPT-2.PRG

------------------ ------------------------------------------------------------


Structure for database : VR-CRATE.DBF
Number of data records :          7
          Last updated : 08/11/93 at 13:44
Field  Field name  Type        Width    Dec    Start      End
    1  AVG_DAYS_L  Numeric         5      1         1        5
    2  CAGE_CODE   Character       5               6       10
    3  CONTR_LINE  Numeric         6              11       16
    4  DELIV_SCOR  Numeric         5      1       17       21
    5  LABTEST_DF  Numeric         5      1       22       26
    6  ON_TIME_PC  Numeric         5      1       27       31
    7  OVERALL_SC  Numeric         5      1       32       36
    8  PACKAGE_DF  Numeric         5      1       37       41
    9  PRODUCT_DF  Numeric         5      1       42       46
   10  QUAL_SCOR   Numeric         5      1       47       51
   11  SCORE_DATE  Date            8              52       59
** Total **                      60


This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\VRCAGE.NDX   (cage_code)

Used by: OPEN_OTHR                 (procedure in C:\DATA\DBASE\RPT-2.PRG)
       : SETUP_FILS                (procedure in C:\DATA\DBASE\RPT-3.PRG)

--------------------------------------------------------------------------------


Structure for database : VRF-RATE.DBF
Number of data records :         10
          Last updated : 08/03/93 at 14:58
Field  Field name  Type        Width    Dec    Start      End
    1  AVG_DAYS_L  Numeric         5      1         1        5
    2  DELIV_SCOR  Numeric         5      1         6       10
    3  FSC_CODE    Character       4               11       14
    4  LABTEST_DF  Numeric         5      1       15       19
    5  ON_TIME_PC  Numeric         5      1       20       24
    6  OVERALL_SC  Numeric         5      1       25       29
    7  PACKAGE_DF  Numeric         5      1       30       34
    8  PRODUCT_DF  Numeric         5      1       35       39
    9  QUAL_SCOR   Numeric         5      1       40       44
   10  SCORE_DATE  Date            8              45       52
** Total **                      53


This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\VRF-RATE.NDX   (fsc_code+dtoc(score_date))

Used by: RPT-4.PRG
       : RPT-10.PRG
       : COMP_BIDS                 (procedure in C:\DATA\DBASE\RPT-2.PRG)
```

```
--------------------------------------------------------------------------------


Structure for database : VENDOR.DBF
Number of data records :         7
          Last updated : 08/11/93 at 13:02
Field  Field name  Type        Width    Dec    Start      End
    1  CAGE_CODE   Character       5              1        5
    2  MANUFACTUR  Character       1              6        6
    3  MFG_FLOOR   Numeric        10              7       16
    4  QUAL_VND_P  Character       1             17       17
    5  TYPE_ORG    Character       1             18       18
    6  TYPE_OWNER  Character       1             19       19
    7  VEND_NAME   Character      30             20       49
** Total **                      50


This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\VENCAGE.NDX  (cage_code)

Used by: RPT-11.PRG
       : SETUP_FILS               (procedure in C:\DATA\DBASE\RPT-3.PRG)

--------------------------------------------------------------------------------


Structure for database : SOLDATA.DBF
Number of data records :         0
          Last updated : 07/08/93 at 15:01
Field  Field name  Type        Width    Dec    Start      End
    1  ABVM_CLAUS  Character       1              1        1
    2  BUY_NO      Character      13              2       14
** Total **                      15


SNAP! did not find any associated index files

Used by: RPT-6.PRG
       : GET_OTHR                 (procedure in C:\DATA\DBASE\RPT-3.PRG)

--------------------------------------------------------------------------------


Structure for database : WORKPERF.DBF
Number of data records :         0
          Last updated : 07/08/93 at 15:26
Field  Field name  Type        Width    Dec    Start      End
    1  AVG_PALT    Numeric         3              1        3
    2  WEEK_NO     Numeric         3              4        6
** Total **                       7


SNAP! did not find any associated index files

Used by: RPT-4.PRG

--------------------------------------------------------------------------------
```

```
Structure for database : DVRS101.DBF
Number of data records :         3
          Last updated : 08/12/93 at 14:28
Field  Field name  Type        Width    Dec    Start      End
    1  CAGE        Character       5              1        5
    2  CONT_LINE   Character      23              6       28
    3  DELIV_DT    Date            8             29       36
    4  FSC         Character       4             37       40
    5  SHIP_DT     Date            8             41       48
** Total **                      49


This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\A101CONT.NDX  (substr(cont_line,1,13))

Used by: RPT-4.PRG
       : SETP_FILS                (procedure in C:\DATA\DBASE\RPT-7.PRG)

--------------------------------------------------------------------------------


Structure for database : DVRS701.DBF
Number of data records :         0
          Last updated : 07/08/93 at 14:35
Field  Field name  Type        Width    Dec    Start      End
    1  CAGE        Character       5              1        5
    2  PIIN_CLIN   Character      23              6       28
    3  CAUS_CD     Character       2             29       30
    4  TEST_DTD    Date            8             31       38
    5  CLOS_DT     Date            8             39       46
    6  DISC_CD     Character       2             47       48
    7  DOC_TYP     Character       1             49       49
    8  FSC         Character       4             50       53
    9  NO_CRIT     Numeric         6             54       59
   10  NO_MAJ      Numeric         6             60       65
   11  NO_MIN      Numeric         6             66       71
   12  NO_TESTED   Numeric         6             72       77
** Total **                      78


This database appears to be associated with index file/tag(s):
       : C:\DATA\DBASE\A701PIIN.NDX  (substr(piin_clin,1,13))

Used by: RPT-4.PRG
       : SETP_FILS                (procedure in C:\DATA\DBASE\RPT-7.PRG)

--------------------------------------------------------------------------------


Structure for database : DVRS00UD.DBF
Number of data records :         0
          Last updated : 07/08/93 at 15:37
Field  Field name  Type        Width    Dec    Start      End
    1  CH_COMP_DT  Date            8              1        8
    2  CH_DECS_DT  Date            8              9       16
    3  CH_RECV_DT  Date            8             17       24
    4  DENIAL_CD   Character       1             25       25
    5  FSC         Character       4             26       29
** Total **                      30


SNAP! did not find any associated index files

Used by: RPT-5.PRG

--------------------------------------------------------------------------------
```

```
Structure for database : DVRS00UQ.DBF
Number of data records :          0
         Last updated : 07/08/93 at 15:37
Field  Field name  Type        Width    Dec    Start      End
    1  CH_COMP_DT  Date            8              1        8
    2  CH_DECS_DT  Date            8              9       16
    3  CH_RECV_DT  Date            8             17       24
    4  DENIAL_CD   Character       1             25       25
    5  FSC         Character       4             26       29
** Total **                      30

SNAP! did not find any associated index files

Used by: RPT-5.PRG

--------------------------------------------------------------------------------

Structure for database : DVRS00PD.DBF
Number of data records :          0
         Last updated : 07/08/93 at 15:37
Field  Field name  Type        Width    Dec    Start      End
    1  CH_COMP_DT  Date            8              1        8
    2  CH_DECS_DT  Date            8              9       16
    3  CH_RECV_DT  Date            8             17       24
    4  DENIAL_CD   Character       1             25       25
    5  FSC         Character       4             26       29
** Total **                      30

SNAP! did not find any associated index files

Used by: RPT-5.PRG

--------------------------------------------------------------------------------

Structure for database : DVRS00PQ.DBF
Number of data records :          0
         Last updated : 07/08/93 at 15:37
Field  Field name  Type        Width    Dec    Start      End
    1  CH_COMP_DT  Date            8              1        8
    2  CH_DECS_DT  Date            8              9       16
    3  CH_RECV_DT  Date            8             17       24
    4  DENIAL_CD   Character       1             25       25
    5  FSC         Character       4             26       29
** Total **                      30

SNAP! did not find any associated index files

Used by: RPT-5.PRG

--------------------------------------------------------------------------------

Structure for database : DVRS00DD.DBF
Number of data records :          0
         Last updated : 07/08/93 at 15:37
Field  Field name  Type        Width    Dec    Start      End
    1  CH_COMP_DT  Date            8              1        8
    2  CH_DECS_DT  Date            8              9       16
    3  CH_RECV_DT  Date            8             17       24
    4  DENIAL_CD   Character       1             25       25
    5  FSC         Character       4             26       29
** Total **                      30

SNAP! did not find any associated index files

Used by: RPT-5.PRG
```

```
--------------------------------------------------------------------------------

Structure for database : DVRS00CQ.DBF
Number of data records :          0
        Last updated : 07/08/93 at 15:37
Field  Field name  Type        Width    Dec    Start      End
    1  CH_COMP_DT  Date            8               1        8
    2  CH_DECS_DT  Date            8               9       16
    3  CH_RECV_DT  Date            8              17       24
    4  DENIAL_CD   Character       1              25       25
    5  FSC         Character       4              26       29
** Total **                       30

SNAP! did not find any associated index files

Used by: RPT-5.PRG

--------------------------------------------------------------------------------

Structure for database : DVRS00CD.DBF
Number of data records :          0
        Last updated : 07/08/93 at 15:37
Field  Field name  Type        Width    Dec    Start      End
    1  CH_COMP_DT  Date            8               1        8
    2  CH_DECS_DT  Date            8               9       16
    3  CH_RECV_DT  Date            8              17       24
    4  DENIAL_CD   Character       1              25       25
    5  FSC         Character       4              26       29
** Total **                       30

SNAP! did not find any associated index files

Used by: RPT-5.PRG

--------------------------------------------------------------------------------

Structure for database : DVRS0A1.DBF
Number of data records :          0
        Last updated : 07/08/93 at 14:30
Field  Field name  Type        Width    Dec    Start      End
    1  CAGE        Character       5               1        5
    2  DELIV_DT    Date            8               6       13
    3  MOD_NUM     Character       6              14       19
    4  CORRECT_DT  Date            8              20       27
    5  NSN         Character      15              28       42
    6  PROJ_ACT_C  Character       2              43       44
    7  QTY_DUE     Numeric         9              45       53
    8  QTY_RECPT   Numeric         9              54       62
    9  QTY_SHIP    Numeric         9              63       71
   10  RSN_DELAY   Character       2              72       73
   11  RECEIPT_DT  Date            8              74       81
   12  SHIP_DT     Date            8              82       89
   13  TERM_CODE   Character       1              90       90
   14  VARIANC_CD  Character       3              91       93
** Total **                       94

SNAP! did not find any associated index files

Used by: RPT-5.PRG

--------------------------------------------------------------------------------
```

```
Structure for database : MUDKEY.DBF
Number of data records :          0
         Last updated : 07/08/93 at 14:55
Field  Field name  Type        Width    Dec    Start      End
    1  BUY_NO      Character      13              1       13
    2  SET_ASIDE   Character       1             14       14
** Total **                      15


SNAP! did not find any associated index files

Used by: RPT-8.PRG
       : RPT-9.PRG

--------------------------------------------------------------------------------


Structure for database : OPENCON.DBF
Number of data records :          0
         Last updated : 07/08/93 at 15:04
Field  Field name  Type        Width    Dec    Start      End
    1  CAGE_CODE   Character       5              1        5
    2  CLIN        Character       6              6       11
    3  CONTRACT_N  Character      17             12       28
    4  DELIV_DT    Date            8             29       36
    5  NSN         Character      15             37       51
** Total **                      52


SNAP! did not find any associated index files

Used by: RPT-11.PRG

--------------------------------------------------------------------------------
```

System: ABVM Decision Support System
Author:
08/12/93 15:17:45
Database Field Summary

Field Name	Type	Len	Dec	Database
ABVM_CLAUS	C	1	0	DWARFDB.DBF
				AWDINFO.DBF
				SOLDATA.DBF
ABVM_D_Q	C	1	0	DWARFDB.DBF
ABVM_P_P	C	1	0	DWARFDB.DBF
ABVM_Q_D	C	1	0	DWARFDB.DBF
AVG_DAYS_L	N	5	1	VR-CRATE.DBF
				VRFCRATE.DBF
				VRF-RATE.DBF
AVG_PALT	N	3	0	WORKPERF.DBF
AWARD_DT	D	8	0	BIDVEN.DBF
AWARD_TOTL	N	16	2	BIDVEN.DBF
BID_ABVM	N	5	1	AWDINFO.DBF
BUS_SIZE	C	1	0	BIDVEN.DBF
BUS_TYPE	C	2	0	BIDVEN.DBF
BUY_NO	C	13	0	DWARFDB.DBF
				SOLDATA.DBF
				BIDVEN.DBF
				MUDKEY.DBF
				AWDINFO.DBF
CAGE	C	5	0	DVRS701.DBF
				DVRS0A1.DBF
				DVRS101.DBF
CAGE_CODE	C	5	0	BIDVEN.DBF
				OPENCON.DBF
				VRFCRATE.DBF
				DWARFDB.DBF
				VENDOR.DBF
				VR-CRATE.DBF
CAUS_CD	C	2	0	DVRS701.DBF
CH_COMP_DT	D	8	0	DVRS00PQ.DBF
				DVRS00PD.DBF
				DVRS00UQ.DBF
				DVRS00CQ.DBF
				DVRS00UD.DBF
				DVRS00DD.DBF
				DVRS00CD.DBF
CH_DECS_DT	D	8	0	DVRS00PD.DBF
				DVRS00UQ.DBF
				DVRS00DD.DBF
				DVRS00UD.DBF
				DVRS00PQ.DBF
				DVRS00CD.DBF
				DVRS00CQ.DBF
CH_RECV_DT	D	8	0	DVRS00UQ.DBF
				DVRS00PQ.DBF
				DVRS00UD.DBF
				DVRS00DD.DBF
				DVRS00CD.DBF
				DVRS00CQ.DBF
				DVRS00PD.DBF
CLIN	C	6	0	OPENCON.DBF
CLOS_DT	D	8	0	DVRS701.DBF
CONTRACT_N	C	17	0	OPENCON.DBF
CONTRACT_N	C	13	0	DWARFDB.DBF
CONTR_LINE	N	6	0	VR-CRATE.DBF
CONT_LINE	C	23	0	DVRS101.DBF
CORRECT_DT	D	8	0	DVRS0A1.DBF
CRIT_QUAL	N	6	0	VRFCRATE.DBF
DELIV_DT	D	8	0	DVRS101.DBF
				DVRS0A1.DBF
				OPENCON.DBF
DELIV_SCOR	N	5	1	VRFCRATE.DBF
				VRF-RATE.DBF

				VR-CRATE.DBF
				BIDVEN.DBF
DENIAL_CD	C	1	0	DVRS00CD.DBF
				DVRS00CQ.DBF
				DVRS00DD.DBF
				DVRS00UD.DBF
				DVRS00UQ.DBF
				DVRS00PD.DBF
				DVRS00PQ.DBF
DIFFERENTL	N	16	2	DWARFDB.DBF
				AWDINFO.DBF
DISC_CD	C	2	0	DVRS701.DBF
DLVY_WGT	N	3	0	BIDVEN.DBF
DOC_TYP	C	1	0	DVRS701.DBF
DVRS_RATIN	N	5	1	BIDVEN.DBF
EXTEND_BID	N	16	2	BIDVEN.DBF
FSC	C	4	0	DVRS00CQ.DBF
				DVRS101.DBF
				DVRS00CD.DBF
				AWDINFO.DBF
				DVRS00DD.DBF
				DVRS701.DBF
				DVRS00PD.DBF
				DVRS00UQ.DBF
				DVRS00PQ.DBF
				DVRS00UD.DBF
FSC_CODE	C	4	0	VRF-RATE.DBF
				VRFCRATE.DBF
LABTEST_DF	N	5	1	VRF-RATE.DBF
				VR-CRATE.DBF
				VRFCRATE.DBF
MANUFACTUR	C	1	0	VENDOR.DBF
MFG_FLOOR	N	10	0	VENDOR.DBF
MOD_NUM	C	6	0	DVRS0A1.DBF
NO_CRIT	N	6	0	DVRS701.DBF
NO_MAJ	N	6	0	DVRS701.DBF
NO_MIN	N	6	0	DVRS701.DBF
NO_TESTED	N	6	0	DVRS701.DBF
NSN	C	15	0	DVRS0A1.DBF
				OPENCON.DBF
				DWARFDB.DBF
ON_TIME_PC	N	5	1	VR-CRATE.DBF
				VRF-RATE.DBF
				VRFCRATE.DBF
OVERALL_SC	N	5	1	VR-CRATE.DBF
				VRFCRATE.DBF
				VRF-RATE.DBF
PACKAGE_DF	N	5	1	VRF-RATE.DBF
				VR-CRATE.DBF
				VRFCRATE.DBF
PCT_CLINS	N	5	1	VRFCRATE.DBF
PIIN_CLIN	C	23	0	DVRS701.DBF
PRODUCT_DF	N	5	1	VRF-RATE.DBF
				VR-CRATE.DBF
				VRFCRATE.DBF
PROJ_ACT_C	C	2	0	DVRS0A1.DBF
QTY_DUE	N	9	0	DVRS0A1.DBF
QTY_RECPT	N	9	0	DVRS0A1.DBF
QTY_SHIP	N	9	0	DVRS0A1.DBF
QUAL_SCOR	N	5	1	BIDVEN.DBF
				VR-CRATE.DBF
				VRFCRATE.DBF
				VRF-RATE.DBF
QUAL_VND_P	C	1	0	VENDOR.DBF
QUAL_WGT	N	3	0	BIDVEN.DBF
RECEIPT_DT	D	8	0	DVRS0A1.DBF
RSN_DELAY	C	2	0	DVRS0A1.DBF
RTD_BIDS	N	5	0	AWDINFO.DBF
SCORE_DATE	D	8	0	VRF-RATE.DBF
				VR-CRATE.DBF
SCORE_DT	D	8	0	VRFCRATE.DBF
SET_ASIDE	C	1	0	MUDKEY.DBF
SHIP_DT	D	8	0	DVRS0A1.DBF
				DVRS101.DBF

TERM_CODE	C	1	0	DVRS0A1.DBF
TEST_DTD	D	8	0	DVRS701.DBF
TOTAL_COST	N	16	2	AWDINFO.DBF
				DWARFDB.DBF
TOT_BIDS	N	5	0	AWDINFO.DBF
TYPE_ORG	C	1	0	VENDOR.DBF
TYPE_OWNER	C	1	0	VENDOR.DBF
VARIANC_CD	C	3	0	DVRS0A1.DBF
VEND_NAME	C	30	0	VENDOR.DBF
WEEK_NO	N	3	0	WORKPERF.DBF

```
*:**************************************************************************
*:
*: Procedure file: C:\DATA\DBASE\MAIN_MNU.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 08/12/93     10:25
*:
*:  Procs & Fncts: BAR_DEF
*:               : BARPOP
*:
*:          Calls: BAR_DEF            (procedure in MAIN_MNU.PRG)
*:               : BARPOP             (procedure in MAIN_MNU.PRG)
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:**************************************************************************
SET TALK OFF
SET ECHO OFF
CLEAR
DO bar_def
ON SELECTION POPUP main_mnu DO barpop
ACTIVATE POPUP main_mnu

*!**************************************************************************
*!
*!      Procedure: BAR_DEF
*!
*!      Called by: MAIN_MNU.PRG
*!
*!**************************************************************************
PROCEDURE bar_def
DEFINE POPUP main_mnu FROM 4,20 TO 19,70;
   MESSAGE " Press number of menu choice, or highlight and press <Enter>"
DEFINE BAR 1 OF main_mnu PROMPT "     == ABVM DSS MAIN MENU ==" SKIP
DEFINE BAR 2 OF main_mnu PROMPT "1 - Summary ABVM Application Statistics"
DEFINE BAR 3 OF main_mnu PROMPT "2 - ABVM Component Score Statistics"
DEFINE BAR 4 OF main_mnu PROMPT "3 - Quality Vendor Program Information"
DEFINE BAR 5 OF main_mnu PROMPT "4 - ABVM Implementation Benefits Indicators"
DEFINE BAR 6 OF main_mnu PROMPT "5 - Challenge Statistics"
DEFINE BAR 7 OF main_mnu PROMPT "6 - ABVM Buyer Performance"
DEFINE BAR 8 OF main_mnu PROMPT "7 - ABVM Program Award Performance"
DEFINE BAR 9 OF main_mnu PROMPT "8 - Center Statistics by Vendor Type"
DEFINE BAR 10 OF main_mnu PROMPT "9 - Center Statistics by Solicitation Factors"
DEFINE BAR 11 OF main_mnu PROMPT "A - ABVM Score Trends"
DEFINE BAR 12 OF main_mnu PROMPT "B - Near-Term Deliveries on Differential Awards"
DEFINE BAR 13 OF main_mnu PROMPT "X - EXIT FROM PROGRAM"
RETURN

*!**************************************************************************
*!
*!      Procedure: BARPOP
*!
*!      Called by: MAIN_MNU.PRG
*!
*!          Calls: RPT-1.PRG
*!               : RPT-2.PRG
*!               : RPT-3.PRG
*!               : RPT-4.PRG
*!               : RPT-5.PRG
*!               : RPT-6.PRG
*!               : RPT-7.PRG
*!               : RPT-8.PRG
*!               : RPT-9.PRG
*!               : RPT-10.PRG
*!               : RPT-11.PRG
*!
*!**************************************************************************
PROCEDURE barpop
DO CASE
CASE BAR() = 2
   CLEAR
   DO rpt-1
   WAIT
```

```
   CLEAR
CASE BAR() = 3
   CLEAR
   DO rpt-2
   WAIT
   CLEAR
CASE BAR() = 4
   CLEAR
   DO rpt-3
   WAIT
   CLEAR
CASE BAR() = 5
   CLEAR
   DO rpt-4
   WAIT
   CLEAR
CASE BAR() = 6
   CLEAR
   DO rpt-5
   WAIT
   CLEAR
CASE BAR() = 7
   CLEAR
   DO rpt-6
   WAIT
   CLEAR
CASE BAR() = 8
   CLEAR
   DO rpt-7
   WAIT
   CLEAR
CASE BAR() = 9
   CLEAR
   DO rpt-8
   WAIT
   CLEAR
CASE BAR() = 10
   CLEAR
   DO rpt-9
   WAIT
   CLEAR
CASE BAR() = 11
   CLEAR
   DO rpt-10
   WAIT
   CLEAR
CASE BAR() = 12
   CLEAR
   DO rpt-11
   WAIT
   CLEAR
CASE BAR() = 13
   DEACTIVATE POPUP
   CLEAR ALL
   RETURN
ENDCASE
RETURN
*: EOF: MAIN_MNU.PRG

-----

*:**********************************************************************************
*:
*: Procedure file: C:\DATA\DBASE\RPT-1.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 08/03/93     14:12
*:
*:  Procs & Fncts: COMP_DIFF
*:               : FSC_STATS
*:               : TAB_AWDS
*:               : RPT1_HEAD
```

```
*:               : RPT1_FSC
*:               : DSC_TAB
*:               : RPT1_DSC
*:
*:         Set by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:          Calls: COMP_DIFF           (procedure in RPT-1.PRG)
*:               : FSC_STATS           (procedure in RPT-1.PRG)
*:
*:           Uses: DWARFDB.DBF
*:               : BIDVEN.DBF
*:
*:        Indexes: AWDBUY.NDX
*:               : BIDBUY.NDX
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:*********************************************************************************
*  rpt-1.prg
*
*  program to compute summary ABVM application statistics (report 1)
*
*  28 jul 93
*
USE dwarfdb
ERASE awdbuy.ndx
INDEX ON buy_no TO awdbuy
USE bidven IN 2
SELECT 2
ERASE bidbuy.ndx
INDEX ON buy_no TO bidbuy
SELECT 1
SET RELATION TO buy_no INTO bidven
SET SKIP TO bidven
DO comp_diff
DO fsc_stats
CLOSE DATABASES
RETURN

*!*********************************************************************************
*!
*!      Procedure: COMP_DIFF
*!
*!      Called by: RPT-1.PRG
*!
*!*********************************************************************************
PROCEDURE comp_diff
*
*  procedure to compute the differential paid (if any) for each award
*
GOTO TOP
SCAN
   SELECT 2
   mbuy = buy_no
   low_bid = bidven->extend_bid
   SCAN WHILE mbuy = buy_no
      new_low = MIN(low_bid, bidven->extend_bid)
      low_bid = new_low
   ENDSCAN
   SELECT 1
   DIFF = dwarfdb->total_cost - low_bid
   diff_pct = 100.0 * DIFF / total_cost
   *  ? mbuy, low_bid, diff, diff_pct
   REPLACE differentl WITH DIFF
ENDSCAN
RETURN

*!*********************************************************************************
*!
*!      Procedure: FSC_STATS
*!
*!      Called by: RPT-1.PRG
*!
*!          Calls: RPT1_HEAD           (procedure in RPT-1.PRG)
*!               : TAB_AWDS            (procedure in RPT-1.PRG)
```

```
*!               : RPT1_FSC           (procedure in RPT-1.PRG)
*!               : RPT1_DSC           (procedure in RPT-1.PRG)
*!
*!           Uses: DWARFDB.DBF
*!
*!        Indexes: AWDNSN.NDX
*!
*!*********************************************************************************
PROCEDURE fsc_stats
*
*  procedure to compute statistics by federal supply class (FSC)
*
CLOSE DATABASES
USE dwarfdb
ERASE awdnsn.ndx
INDEX ON nsn TO awdnsn
GOTO TOP
DO rpt1_head
d_awds = 0
d_diff = 0
d_abvm = 0
dt_diff = 0.0
d_cost = 0.0
d_pabvm = 0.0
d_pdiff = 0.0
d_adiff = 0.0
d_apct = 0.0
numb_awds = 0
numb_diff = 0
numb_abvm = 0
total_diff = 0.0
ttl_cost = 0.0
pct_abvm = 0.0
pct_diff = 0.0
ave_diff = 0.0
ave_pct = 0.0
fsc = SUBSTR(nsn,1,4)
SCAN
   SCAN WHILE fsc=SUBSTR(nsn,1,4)
      DO tab_awds
   ENDSCAN
   DO rpt1_fsc
   *
   *  must reset statistics for next FSC
   *
   fsc = SUBSTR(nsn,1,4)
   numb_awds = 0
   numb_diff = 0
   numb_abvm = 0
   total_diff = 0.0
   ttl_cost = 0.0
   pct_abvm = 0.0
   pct_diff = 0.0
   ave_diff = 0.0
   ave_pct = 0.0
   DO tab_awds
ENDSCAN
IF fsc <> '    '
   DO rpt1_fsc
ENDIF
DO rpt1_dsc
RETURN

*!*********************************************************************************
*!
*!      Procedure: TAB_AWDS
*!
*!      Called by: FSC_STATS          (procedure in RPT-1.PRG)
*!
*!*********************************************************************************
PROCEDURE tab_awds
*
*  tabulates award statistics
*
```

```
numb_awds = numb_awds + 1
IF abvm_clause = 'y'
   numb_abvm = numb_abvm + 1
ENDIF
IF differentl > 0.0
   numb_diff = numb_diff + 1
   total_diff = total_diff + differentl
   ttl_cost = ttl_cost + total_cost
ENDIF
RETURN

*!*************************************************************************
*!
*!      Procedure: RPT1_HEAD
*!
*!      Called by: FSC_STATS          (procedure in RPT-1.PRG)
*!
*!*************************************************************************
PROCEDURE rpt1_head
*
*  prints output heading
*
?
? "Report 1 - Summary ABVM Application Statistics"
?
?
? "Number" AT 8, "Percent" AT 18, "Percent" AT 28, "Average" AT 38, ;
   "Average" AT 48
? "FSC" AT 1, "Awards" AT 8, "ABVM" AT 19, "w/ Diff" AT 28, "Diff ($)" AT 38, ;
   "Diff (pct)" AT 48
?
RETURN

*!*************************************************************************
*!
*!      Procedure: RPT1_FSC
*!
*!      Called by: FSC_STATS          (procedure in RPT-1.PRG)
*!
*!          Calls: DSC_TAB            (procedure in RPT-1.PRG)
*!
*!*************************************************************************
PROCEDURE rpt1_fsc
*
*  produces output statistics for each FSC
*
pct_abvm = 100.0 * numb_abvm / numb_awds
pct_diff = 100.0 * numb_diff / numb_awds
IF numb_diff > 0
   ave_diff = total_diff / numb_diff
   ave_pct = 100.0 * total_diff / ttl_cost
ENDIF
?
?? fsc PICTURE "XXXX" AT 1
?? numb_awds PICTURE "99999" AT 9
?? pct_abvm PICTURE "999.99" AT 19
?? pct_diff PICTURE "999.99" AT 29
?? ave_diff PICTURE "99999.99" AT 38
?? ave_pct PICTURE "999.99" AT 49
DO dsc_tab
* ? fsc, numb_awds, pct_abvm, pct_diff, ave_diff
RETURN

*!*************************************************************************
*!
*!      Procedure: DSC_TAB
*!
*!      Called by: RPT1_FSC           (procedure in RPT-1.PRG)
*!
*!*************************************************************************
PROCEDURE dsc_tab
*
*  tabulates statistics at dsc level
*
```

```
d_abvm = d_abvm + numb_abvm
d_awds = d_awds + numb_awds
d_diff = d_diff + numb_diff
dt_diff = dt_diff + total_diff
d_cost = d_cost + ttl_cost
RETURN

*!*************************************************************************
*!
*!      Procedure: RPT1_DSC
*!
*!      Called by: FSC_STATS          (procedure in RPT-1.PRG)
*!
*!*************************************************************************
PROCEDURE rpt1_dsc
*
*  computes and prints dsc summary statistics
*
d_pabvm = 100.0 * d_abvm / d_awds
d_pdiff = 100.0 * d_diff / d_awds
IF d_diff > 0
   d_adiff = dt_diff / d_diff
   d_apct = 100.0 * dt_diff / d_cost
ENDIF
?
?
?? "DSC" AT 1
?? d_awds PICTURE "99999" AT 9
?? d_pabvm PICTURE "999.99" AT 19
?? d_pdiff PICTURE "999.99" AT 29
?? d_adiff PICTURE "99999.99" AT 38
?? d_apct PICTURE "999.99" AT 49
RETURN
*: EOF: RPT-1.PRG

-----

*:*********************** ************************************************
*:
*: Procedure file: C:\DATA\DBASE\RPT-2.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 08/03/93     15:15
*:
*:  Procs & Fncts: RPT2_HDR
*:               : COMP_AWDS
*:               : COMP_BIDS
*:               : OPEN_OTHR
*:               : RPT2_TOTLS
*:               : WRT_RCD
*:               : PROC_INFO
*:               : BID_HDR
*:
*:         Set by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:          Calls: RPT2_HDR            (procedure in RPT-2.PRG)
*:               : COMP_AWDS           (procedure in RPT-2.PRG)
*:               : COMP_BIDS           (procedure in RPT-2.PRG)
*:               : OPEN_OTHR           (procedure in RPT-2.PRG)
*:
*:           Uses: VRFCRATE.DBF
*:               : DWARFDB.DBF
*:               : AWDINFO.DBF
*:
*:        Indexes: VRFCAGE.NDX
*:               : AWDKEY.NDX
*:               : AWDCAGE.NDX
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:*************************************************************************
*  rpt-2.prg
*
```

```
*  program to compute ABVM component score statistics (report 2)
*
*  2 aug 93
*
USE vrfcrate
ERASE vrfcage.ndx
INDEX ON cage_code+fsc_code TO vrfcage
USE dwarfdb IN 2
USE awdinfo IN 5
SELECT 5
ERASE awdkey.ndx
INDEX ON buy_no TO awdkey
SELECT 2
ERASE awdcage.ndx
INDEX ON cage_code+SUBSTR(nsn,1,4) TO awdcage
SET RELATION TO cage_code+SUBSTR(nsn,1,4) INTO vrfcrate
SET SKIP TO vrfcrate
DO rpt2_hdr
DO comp_awds
?
WAIT
DO comp_bids
DO open_othr
CLOSE DATABASES
RETURN

*!*************************************************************************
*!
*!      Procedure: RPT2_HDR
*!
*!      Called by: RPT-2.PRG
*!
*!*************************************************************************
PROCEDURE rpt2_hdr
CLEAR
?
? "Report 2 - ABVM Component Score Statistics"
?
RETURN

*!*************************************************************************
*!
*!      Procedure: COMP_AWDS
*!
*!      Called by: RPT-2.PRG
*!
*!          Calls: RPT2_TOTLS         (procedure in RPT-2.PRG)
*!
*!*************************************************************************
PROCEDURE comp_awds
**********
*  must ensure that score is pulled for appropriate fsc
**********
SELECT 2
GOTO TOP
numb_abvm = 0
numb_non = 0
numb_tot = 0
a_overall = 0.0
a_ave = 0.0
a_deliv = 0.0
a_qual = 0.0
n_overall = 0.0
n_ave = 0.0
n_deliv = 0.0
n_qual = 0.0
x_overall = 0.0
x_ave = 0.0
x_deliv = 0.0
x_qual = 0.0
ttl_awds = 0
SCAN
   ttl_awds = ttl_awds + 1
   mfsc=SUBSTR(nsn,1,4)
```

```
   mcage=cage_code
   *  ? mfsc, mcage
   mabvm=abvm_claus
   SELECT 1
   SEEK mcage+mfsc
   IF FOUND()
      *    ? vrfcrate->overall_sc
      IF mabvm = 'y'
         numb_abvm = numb_abvm + 1
         a_overall = a_overall + vrfcrate->overall_sc
         a_deliv = a_deliv + vrfcrate->deliv_scor
         a_qual = a_qual + vrfcrate->qual_scor
      ELSE
         numb_non = numb_non + 1
         n_overall = n_overall + vrfcrate->overall_sc
         n_deliv = n_deliv + vrfcrate->deliv_scor
         n_qual = n_qual + vrfcrate->qual_scor
      ENDIF
   ELSE
      *    ? "unable to locate score"
   ENDIF
   SELECT 2
ENDSCAN
a_ave = a_overall / numb_abvm
n_ave = n_overall / numb_non
a_deliv = a_deliv / numb_abvm
a_qual = a_qual / numb_abvm
n_deliv = n_deliv / numb_non
n_qual = n_qual / numb_non
numb_tot = numb_abvm + numb_non
x_ave = (a_overall + n_overall) / numb_tot
x_deliv = ((a_deliv * numb_abvm) + (n_deliv * numb_non)) / numb_tot
x_qual = ((a_qual * numb_abvm) + (n_qual * numb_non)) / numb_tot
DO rpt2_totls
RETURN

*!*****************************************************************************
*!
*!      Procedure: RPT2_TOTLS
*!
*!      Called by: COMP_AWDS          (procedure in RPT-2.PRG)
*!
*!*****************************************************************************
PROCEDURE rpt2_totls
?
? "Part 2.1 - Awardee ABVM Statistics"
?
? "total awards = "
?? ttl_awds PICTURE "99999" AT 18
?
? "overall" AT 26, "delivery" AT 35, "quality" AT 46
? "number" AT 18, "average" AT 26, "average" AT 36, "average" AT 46
?
? "    abvm awards" AT 1
?? numb_abvm PICTURE "9999" AT 20, a_ave PICTURE "999.9" AT 27, ;
   a_deliv PICTURE "999.9" AT 37, a_qual PICTURE "999.9" AT 47
? "non-abvm awards" AT 1
?? numb_non PICTURE "9999" AT 20, n_ave PICTURE "999.9" AT 27, ;
   n_deliv PICTURE "999.9" AT 37, n_qual PICTURE "999.9" AT 47
?
? "     all awards" AT 1
?? numb_tot PICTURE "9999" AT 20, x_ave PICTURE "999.9" AT 27, ;
   x_deliv PICTURE "999.9" AT 37, x_qual PICTURE "999.9" AT 47
RETURN

*!*****************************************************************************
*!
*!      Procedure: OPEN_OTHR
*!
*!      Called by: RPT-2.PRG
*!
*!           Uses: BIDVEN.DBF
*!               : VR-CRATE.DBF
*!
```

```
*!*************************************************************************
PROCEDURE open_othr
*
*  developmental procedure to open other database files
*
USE bidven IN 3
USE vr-crate IN 4
RETURN

*!*************************************************************************
*!
*!      Procedure: COMP_BIDS
*!
*!      Called by: RPT-2.PRG
*!
*!          Calls: WRT_RCD             (procedure in RPT-2.PRG)
*!               : PROC_INFO           (procedure in RPT-2.PRG)
*!
*!           Uses: VRF-RATE.DBF
*!               : BIDVEN.DBF
*!
*!        Indexes: AWDBUY.NDX
*!               : BIDBUY.NDX
*!
*!*************************************************************************
PROCEDURE comp_bids
*
*  procedure to compute the statistics for bids for each award
*
SELECT 2
ERASE awdbuy.ndx
INDEX ON buy_no TO awdbuy
*
USE vrf-rate IN 6
USE bidven IN 3
SELECT 3
ERASE bidbuy.ndx
INDEX ON buy_no TO bidbuy
GOTO TOP

SCAN

   *   store the buy number to a memory variable

   mbuy=buy_no

   *   fetch the corresponding fsc for the award

   SELECT 2
   GOTO TOP
   SEEK mbuy
   IF FOUND()
      mfsc=SUBSTR(nsn,1,4)
   ELSE
      mfsc='    '
   ENDIF
   *
   SELECT 3
   numb_bid = 0
   f_overall = 0.0
   z_overall = 0.0

   bids = 0
   ns_bids = 0
   SCAN WHILE mbuy=buy_no
      mcage=cage_code
      bids = bids + 1

      SELECT 1
      GOTO TOP
      SEEK mcage+mfsc
      IF FOUND()
         numb_bid = numb_bid + 1
         f_overall = f_overall + vrfcrate->overall_sc
```

```
      ENDIF
      SELECT 3
   ENDSCAN

   IF numb_bid > 0
      z_overall = f_overall / numb_bid
   ELSE
      z_overall = 0.0
   ENDIF
   ns_bids = bids - numb_bid
   mrecno = RECNO()
   DO wrt_rcd
   GOTO mrecno-1
   mbuy = buy_no
ENDSCAN
DO proc_info
RETURN

*!*************************************************************************
*!
*!      Procedure: WRT_RCD
*!
*!      Called by: COMP_BIDS          (procedure in RPT-2.PRG)
*!
*!*************************************************************************
PROCEDURE wrt_rcd
SELECT 5
IF .NOT. SEEK(mbuy)
   APPEND BLANK
   REPLACE buy_no WITH mbuy
ELSE
   SEEK(mbuy)
ENDIF
REPLACE fsc WITH mfsc
REPLACE tot_bids WITH bids
REPLACE rtd_bids WITH numb_bid
REPLACE bid_abvm WITH z_overall
xrecno = RECNO()
SELECT 2
SEEK(mbuy)
IF FOUND()
   xclause = dwarfdb->abvm_claus
   xdiff = dwarfdb->differentl
   xcost = dwarfdb->total_cost
ELSE
   xclause = ' '
   xdiff = 0.0
   xcost = 0.0
ENDIF
SELECT 5
GOTO xrecno
REPLACE abvm_claus WITH xclause
REPLACE differentl WITH xdiff
REPLACE total_cost WITH xcost
SELECT 3
RETURN

*!*************************************************************************
*!
*!      Procedure: PROC_INFO
*!
*!      Called by: COMP_BIDS          (procedure in RPT-2.PRG)
*!
*!          Calls: BID_HDR            (procedure in RPT-2.PRG)
*!
*!        Indexes: ABVMFSC.NDX
*!               : AWDFSC.NDX
*!
*!*************************************************************************
PROCEDURE proc_info
*
*  processes award/bid information
*
?
```

```
*
SELECT 6
ERASE abvmfsc.ndx
INDEX ON fsc_code TO abvmfsc
*
SELECT 5
ERASE awdfsc.ndx
INDEX ON fsc TO awdfsc
DO bid_hdr
SCAN
   numb_awds = 0
   numb_rtd = 0
   numb_bids = 0
   sum_score = 0.0
   numb_abvm = 0
   numb_non = 0
   rtd_abvm = 0
   rtd_nona = 0
   sum_a_rtd = 0.0
   sum_n_rtd = 0.0
   tot_abids = 0
   tot_nbids = 0

   xfsc = fsc
   SCAN WHILE xfsc=fsc
      numb_awds = numb_awds + 1
      numb_rtd = numb_rtd + awdinfo->rtd_bids
      numb_bids = numb_bids + awdinfo->tot_bids
      sum_score = sum_score + (rtd_bids * bid_abvm)
      IF abvm_claus = 'y'
         numb_abvm = numb_abvm + 1
         rtd_abvm = rtd_abvm + awdinfo->rtd_bids
         tot_abids = tot_abids + awdinfo->tot_bids
         sum_a_rtd = sum_a_rtd + (rtd_bids * bid_abvm)
      ELSE
         numb_non = numb_non + 1
         rtd_nona = rtd_nona + awdinfo->rtd_bids
         tot_nbids = tot_nbids + awdinfo->tot_bids
         sum_n_rtd = sum_n_rtd + (rtd_bids * bid_abvm)
      ENDIF
   ENDSCAN
   IF rtd_abvm > 0
      sum_a_rtd = sum_a_rtd / rtd_abvm
   ELSE
      sum_a_rtd = 0.0
   ENDIF
   IF rtd_nona > 0
      sum_n_rtd = sum_n_rtd / rtd_nona
   ELSE
      sum_n_rtd = 0.0
   ENDIF
   ?
   ?? xfsc PICTURE "XXXX" AT 1
   ?? numb_awds PICTURE "99999" AT 9
   ?? numb_bids PICTURE "99999" AT 18
   ?? numb_rtd PICTURE "99999" AT 28
   IF numb_rtd > 0
      sum_score = sum_score / numb_rtd
   ELSE
      sum_score = 0.0
   ENDIF
   ?? sum_score PICTURE "999.9" AT 38
   ?? numb_abvm PICTURE "99999" AT 47
   ?? sum_a_rtd PICTURE "999.9" AT 55
   ?? sum_n_rtd PICTURE "999.9" AT 63
   xrecno=RECNO()
   *
   *  must modify somehow to find the fsc score
   *  with the most recent date, since there may
   *  be numerous fsc scores archived
   *
   SELECT 6
   *
   *  probably, use of a scan would help,
```

```
   *  but the file must be indexed differently
   *
   SEEK(xfsc)
   IF FOUND()
      abvm_fsc = overall_sc
   ELSE
      abvm_fsc = 0.0
   ENDIF
   ?? abvm_fsc PICTURE "999.9" AT 71
   *
   SELECT 5
   GOTO xrecno-1
ENDSCAN
SELECT 3
RETURN

*!*************************************************************************
*!
*!      Procedure: BID_HDR
*!
*!      Called by: PROC_INFO          (procedure in RPT-2.PRG)
*!
*!*************************************************************************
PROCEDURE bid_hdr
?
? "Part 2.2 - Bidder ABVM Statistics"
?
? "score" AT 55, "score" AT 63, "FSC" AT 72
? "number" AT 9, "total" AT 18, "rated" AT 28, "average" AT 37, "# ABVM" AT 47, ;
   "ABVM" AT 56, "non-ABVM" AT 62, "ABVM" AT 72
? "FSC" AT 1, "awards" AT 9, "bids" AT 19, "bids" AT 29, "score bid" AT 36, ;
   "awards" AT 47, "bids" AT 56, "bids" AT 64, "score" AT 71
?
RETURN
*: EOF: RPT-2.PRG

-----

*:*************************************************************************
*:
*: Procedure file: C:\DATA\DBASE\RPT-3.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 08/12/93     10:20
*:
*:  Procs & Fncts: SETUP_FILS
*:               : ADD_AWDS
*:               : GET_OTHR
*:               : SUM_FSC
*:               : SUM_CAGE
*:               : OUT_INFO
*:               : GET_ABVM
*:               : GET_F_ABVM
*:
*:         Set by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:          Calls: SETUP_FILS          (procedure in RPT-3.PRG)
*:               : ADD_AWDS            (procedure in RPT-3.PRG)
*:               : GET_OTHR            (procedure in RPT-3.PRG)
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:*************************************************************************
*  rpt-3.prg
*
*  program to compute QVP information (report 3)
*
*
DO setup_fils
SELECT 1
DO add_awds
DO get_othr
CLOSE DATABASES
```

```
RETURN

*!*************************************************************************
*!
*!      Procedure: SETUP_FILS
*!
*!      Called by: RPT-3.PRG
*!
*!           Uses: VENDOR.DBF
*!               : DWARFDB.DBF
 !               : VR-CRATE.DBF
*!               : VRFCRATE.DBF
*!
*!        Indexes: VENCAGE.NDX
*!               : AWDCAGE.NDX
*!               : VRCAGE.NDX
*!               : VRFCAGE.NDX
*!
*!*************************************************************************
PROCEDURE setup_fils
*
*  sets up database files to be used for this report
*
*  area 1 ==> vendor
*  area 2 ==> dwarfdb
*  area 3 ==> vr-crate
*  area 4 ==> vrfcrate
*
USE vendor
ERASE vencage.ndx
INDEX ON cage_code TO vencage
*
USE dwarfdb IN 2
SELECT 2
ERASE awdcage.ndx
INDEX ON cage_code+SUBSTR(nsn,1,4) TO awdcage
*
USE vr-crate IN 3
SELECT 3
ERASE vrcage.ndx
INDEX ON cage_code TO vrcage
*
USE vrfcrate IN 4
SELECT 4
ERASE vrfcage.ndx
INDEX ON cage_code+fsc_code TO vrfcage
*
RETURN


*!*************************************************************************
*!
*!      Procedure: ADD_AWDS
*!
*!      Called by: RPT-3.PRG
*!
*!          Calls: SUM_FSC            (procedure in RPT-3.PRG)
*!               : SUM_CAGE           (procedure in RPT-3.PRG)
*!               : GET_F_ABVM         (procedure in RPT-3.PRG)
*!               : OUT_INFO           (procedure in RPT-3.PRG)
*!
*!*************************************************************************
PROCEDURE add_awds
*
*  procedure to aggregate award information by vendor
*
SELECT 1
GOTO TOP
?
? "Report 3 - Quality Vendor Program Information"
?
? "No." AT 29, "Award" AT 36, "Diff" AT 46, "Diff" AT 55, "Bus" AT 64, ;
   "Overall" AT 72
? "CAGE" AT 1, "Vendor Name" AT 7, "Awds" AT 28, "Dollars" AT 36, ;
```

```
   "Awds" AT 46, "Dollars" AT 54, "Sz" AT 65, "Mf" AT 69, "ABVM" AT 73
SCAN
   mcage = cage_code

   *  now use dwarfdb to find appropriate awards for vendor

   SELECT 2
   numb_awds = 0
   numb_diff = 0
   sum_value = 0.0
   sum_diff = 0.0
   SCAN
      IF mcage = cage_code
         *
         *
         mfsc = SUBSTR(nsn,1,4)
         numb_f_awd = 0
         numb_f_dif = 0
         sum_f_valu = 0.0
         sum_f_diff = 0.0
         DO sum_fsc
         DO sum_cage
         *
         SELECT 1
         IF qual_vnd_p = 'y'
            ?
            ? mcage PICTURE "XXXXX" AT 1
            ?? mfsc PICTURE "XXXX" AT 10
            ?? numb_f_awd PICTURE "9999" AT 28
            ?? sum_f_valu PICTURE "9999999.99" AT 33
            ?? numb_f_dif PICTURE "9999" AT 45
            ?? sum_f_diff PICTURE "9999999.99" AT  51
            abvm_sc = 0.0
            DO get_f_abvm
            ?? abvm_sc PICTURE "999.9" AT 73
         ENDIF
         SELECT 2
         *
      ENDIF
   ENDSCAN

   *  go back to vendor file

   SELECT 1
   DO out_info
ENDSCAN
RETURN

*!*************************************************************************
*!
*!      Procedure: SUM_FSC
*!
*!      Called by: ADD_AWDS           (procedure in RPT-3.PRG)
*!
*!*************************************************************************
PROCEDURE sum_fsc
*
*  add statistics for awards of current fsc and cage
*
SCAN FOR (mfsc = SUBSTR(nsn,1,4)) .AND. (mcage = cage_code)
   *
   numb_f_awd = numb_f_awd + 1
   sum_f_valu = sum_f_valu + total_cost
   IF differentl > 0.0
      numb_f_dif = numb_f_dif + 1
      sum_f_diff = sum_f_diff + differentl
   ENDIF
   *
   LASTREC=RECNO()
ENDSCAN
GOTO LASTREC
RETURN

*!*************************************************************************
```

```
*!
*!      Procedure: SUM_CAGE
*!
*!      Called by: ADD_AWDS           (procedure in RPT-3.PRG)
*!
*!*********************************************************************************
PROCEDURE sum_cage
*
*  add fsc statistics for cage code
*
numb_awds = numb_awds + numb_f_awd
sum_value = sum_value + sum_f_valu
IF sum_f_diff > 0.0
   numb_diff = numb_diff + numb_f_dif
   sum_diff = sum_diff + sum_f_diff
ENDIF
RETURN

*!*********************************************************************************
*!
*!      Procedure: OUT_INFO
*!
*!      Called by: ADD_AWDS           (procedure in RPT-3.PRG)
*!
*!          Calls: GET_ABVM           (procedure in RPT-3.PRG)
*!
*!*********************************************************************************
PROCEDURE out_info
*
*  write output for appropriate vendors
*
IF qual_vnd_p = 'y'
   ? " ------------------------------------------------------------------------"
   ? mcage PICTURE "XXXXX" AT 1
   ?? vend_name PICTURE "XXXXXXXXXXXXXXXXXXXX" AT 7
   ?? numb_awds PICTURE "9999" AT 28
   ?? sum_value PICTURE "9999999.99" AT 33
   ?? numb_diff PICTURE "9999" AT 45
   ?? sum_diff PICTURE "9999999.99" AT  51
   ?? type_org PICTURE "X" AT 65
   ?? manufactur PICTURE "X" AT 70
   abvm_sc = 0.0
   DO get_abvm
   ?? abvm_sc PICTURE "999.9" AT 73
ENDIF
?
RETURN

*!*********************************************************************************
*!
*!      Procedure: GET_ABVM
*!
*!      Called by: OUT_INFO           (procedure in RPT-3.PRG)
*!
*!*********************************************************************************
PROCEDURE get_abvm
*
*  fetch abvm rating information
*
SELECT 3
GOTO TOP
SEEK(mcage)
IF FOUND()
   abvm_sc = overall_sc
ELSE
   abvm_sc = 0.0
ENDIF
SELECT 1
RETURN

*!*********************************************************************************
*!
*!      Procedure: GET_F_ABVM
*!
```

```
*!      Called by: ADD_AWDS           (procedure in RPT-3.PRG)
*!
*!*************************************************************************
PROCEDURE get_f_abvm
*
*  fetch abvm fsc rating information
*
SELECT 4
GOTO TOP
SEEK(mcage+mfsc)
IF FOUND()
   abvm_sc = overall_sc
ELSE
   abvm_sc = 0.0
ENDIF
SELECT 1
RETURN

*!*************************************************************************
*!
*!      Procedure: GET_OTHR
*!
*!      Called by: RPT-3.PRG
*!
*!           Uses: BIDVEN.DBF
*!               : SOLDATA.DBF
*!
*!*************************************************************************
PROCEDURE get_othr
*
*  temporary procedure to open other database files
*  to be used by this report
*
CLOSE DATABASES
USE bidven IN 5
USE soldata IN 6
CLOSE DATABASES
RETURN

*: EOF: RPT-3.PRG

-----

*:*************************************************************************
*:
*:        Program: C:\DATA\DBASE\RPT-4.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 07/29/93     14:45
*:
*:      Called by: BARPOP             (procedure in MAIN_MNU.PRG)
*:
*:           Uses: VRF-RATE.DBF
*:               : DWARFDB.DBF
*:               : BIDVEN.DBF
*:               : WORKPERF.DBF
*:               : DVRS101.DBF
*:               : DVRS701.DBF
*:
*:      Documented 08/12/93 at 15:17               SNAP!  version 5.01
*:*************************************************************************
? "in report 4"
USE vrf-rate
USE dwarfdb IN 2
USE bidven IN 3
USE workperf IN 4
USE dvrs101 IN 5
USE dvrs701 IN 6
CLOSE DATABASES
RETURN
*: EOF: RPT-4.PRG
```

```
-----

*:**************************************************************************
*:
*:        Program: C:\DATA\DBASE\RPT-5.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 07/29/93     14:46
*:
*:      Called by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:           Uses: DVRS00UD.DBF
*:               : DVRS00UQ.DBF
*:               : DVRS00PD.DBF
*:               : DVRS00PQ.DBF
*:               : DVRS00DD.DBF
*:               : DVRS00CQ.DBF
*:               : DVRS00CD.DBF
*:               : DVRS0A1.DBF
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:**************************************************************************
? "in report 5"
USE dvrs00ud
USE dvrs00uq IN 2
USE dvrs00pd IN 3
USE dvrs00pq IN 4
USE dvrs00dd IN 5
USE dvrs00cq IN 6
USE dvrs00cd IN 7
USE dvrs0a1 IN 8
CLOSE DATABASES
RETURN
*: EOF: RPT-5.PRG

-----

*:**************************************************************************
*:
*:        Program: C:\DATA\DBASE\RPT-6.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 07/29/93     14:46
*:
*:      Called by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:           Uses: DWARFDB.DBF
*:               : BIDVEN.DBF
*:               : SOLDATA.DBF
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:**************************************************************************
? "in report 6"
USE dwarfdb
USE bidven IN 2
USE soldata IN 3
CLOSE DATABASES
RETURN
*: EOF: RPT-6.PRG

-----

*:**************************************************************************
*:
*: Procedure file: C:\DATA\DBASE\RPT-7.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 08/12/93     15:15
```

```
*:
*:  Procs & Fncts: SETP_FILS
*:               : BRK_FSC
*:               : COMP_LATE
*:               : ADD_LATE
*:               : TAB_CAT
*:               : HDR7_FSC
*:               : OUT_INFO_F
*:
*:         Set by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:          Calls: SETP_FILS           (procedure in RPT-7.PRG)
*:               : BRK_FSC             (procedure in RPT-7.PRG)
*:
*:      Documented 08/12/93 at 15:17               SNAP!  version 5.01
*:*****************************************************************************
*  rpt-7.prg
*
*  program to compute ABVM program award information (report 7)
*
*
CLEAR
DO setp_fils
SELECT 1
DO brk_fsc
CLOSE DATABASES
RETURN

*!*****************************************************************************
*!
*!      Procedure: SETP_FILS
*!
*!      Called by: RPT-7.PRG
*!
*!           Uses: DWARFDB.DBF
*!               : DVRS101.DBF
*!               : DVRS701.DBF
*!
*!        Indexes: AWDCONT.NDX
*!               : A101CONT.NDX
*!               : A701PIIN.NDX
*!
*!*****************************************************************************
PROCEDURE setp_fils
*
*  sets up database files to be used for this report
*
*  area 1 ==> dwarfdb
*  area 2 ==> dvrs101
*  area 3 ==> dvrs701
*
USE dwarfdb
SELECT 1
ERASE awdcont.ndx
INDEX ON SUBSTR(nsn,1,4)+contract_n TO awdcont
*
USE dvrs101 IN 2
SELECT 2
ERASE a101cont.ndx
INDEX ON SUBSTR(cont_line,1,13) TO a101cont
*
USE dvrs701 IN 3
SELECT 3
ERASE a701piin.ndx
INDEX ON SUBSTR(piin_clin,1,13) TO a701piin
*
RETURN


*!*****************************************************************************
*!
*!      Procedure: BRK_FSC
*!
*!      Called by: RPT-7.PRG
```

```
*!
*!          Calls: HDR7_FSC           (procedure in RPT-7.PRG)
*!               : COMP_LATE          (procedure in RPT-7.PRG)
*!               : ADD_LATE           (procedure in RPT-7.PRG)
*!               : OUT_INFO_F         (procedure in RPT-7.PRG)
*!
*!*********************************************************************************
PROCEDURE brk_fsc
*
*  procedure to track performance award information by fsc
*
SELECT 1
GOTO TOP
?
? "Report 7 - ABVM Program Award Performance"
?
SCAN
   mfsc = SUBSTR(nsn,1,4)

   *  reset fsc statistics

   numb_delv = 0
   late_delv = 0
   late_tot = 0
   a_numb = 0
   a_late = 0
   a_days = 0
   d_numb = 0
   d_late = 0
   d_days = 0
   n_numb = 0
   n_late = 0
   n_days = 0

   *  write header for fsc output

   DO hdr7_fsc

   ktype = " "
   SCAN FOR mfsc = SUBSTR(nsn,1,4)
      mcont = contract_n

      IF abvm_claus = "y"
         ktype = "a"
         IF differentl > 0.0
            ktype = "d"
         ENDIF
      ELSE
         ktype = "n"
      ENDIF

      *  now use dvrs101 to find performance for given award

      SELECT 2
      SEEK(mcont)
      IF FOUND()
         *     compute whether delinquent
         late_days = 0
         DO comp_late
         DO add_late
      ELSE
         *     do nothing; move to next contract
      ENDIF
      SELECT 1
      LASTREC = RECNO()
   ENDSCAN
   GOTO LASTREC
   DO out_info_f
ENDSCAN
RETURN

*!*********************************************************************************
*!
*!       Procedure: COMP_LATE
```

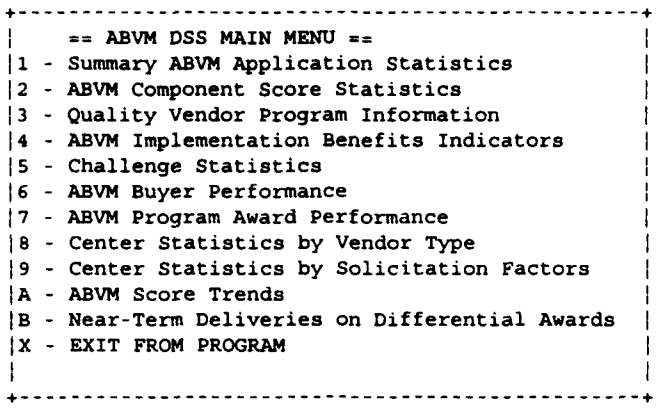
```
*!
*!      Called by: BRK_FSC              (procedure in RPT-7.PRG)
*!
*!*************************************************************************
PROCEDURE comp_late
*
*  determine if particular contract was delinquent
*
late_days = ship_dt - deliv_dt
IF late_days < 0
   late_days = 0
ENDIF
RETURN

*!*************************************************************************
*!
*!      Procedure: ADD_LATE
*!
*!      Called by: BRK_FSC              (procedure in RPT-7.PRG)
*!
*!          Calls: TAB_CAT              (procedure in RPT-7.PRG)
*!
*!*************************************************************************
PROCEDURE add_late
*
*  aggregate statistics on late deliveries for fsc
*
IF ship_dt > CTOD("01/01/70")
   numb_delv = numb_delv + 1
   IF late_days > 0
      late_delv = late_delv + 1
      late_tot = late_tot + late_days
   ENDIF
   DO tab_cat
ENDIF
RETURN

*!*************************************************************************
*!
*!      Procedure: TAB_CAT
*!
*!      Called by: ADD_LATE             (procedure in RPT-7.PRG)
*!
*!*************************************************************************
PROCEDURE tab_cat
*
*  aggregate statistics by abvm category
*
DO CASE

   *  abvm, no differential

CASE ktype = "a"
   a_numb = a_numb + 1
   IF late_days > 0
      a_late = a_late + 1
      a_days = a_days + late_days
   ENDIF

   *  abvm, with differential

CASE ktype = "d"
   d_numb = d_numb + 1
   IF late_days > 0
      d_late = d_late + 1
      d_days = d_days + late_days
   ENDIF

   *  non-abvm

CASE ktype = "n"
   n_numb = n_numb + 1
   IF late_days > 0
      n_late = n_late + 1
```

```
      n_days = n_days + late_days
   ENDIF
ENDCASE
RETURN


*!*************************************************************************
*!
*!      Procedure: HDR7_FSC
*!
*!      Called by: BRK_FSC              (procedure in RPT-7.PRG)
*!
*!*************************************************************************
PROCEDURE hdr7_fsc
?
? "ABVM" AT 19, "ABVM" AT 29, "Non" AT 39
? "FSC = " AT 1
?? mfsc PICTURE "XXXX" AT 10
?? "Diff" AT 19, "No Diff" AT 28, "ABVM" AT 39, "Total" AT 48
?
RETURN


*!*************************************************************************
*!
*!      Procedure: OUT_INFO_F
*!
*!      Called by: BRK_FSC              (procedure in RPT-7.PRG)
*!
*!*************************************************************************
PROCEDURE out_info_f
*
*  write output for fsc
*
? "% delinquent" AT 1
d_rate = 100.0 * d_late / d_numb
?? d_rate PICTURE "999.9" AT 18
a_rate = 100.0 * a_late / a_numb
?? a_rate PICTURE "999.9" AT 29
n_rate = 100.0 * n_late / n_numb
?? n_rate PICTURE "999.9" AT 39
late_rate = 100.0 * late_delv / numb_delv
?? late_rate PICTURE "999.9" AT 48
? "avg days late" AT 1
d_ave = d_days / d_late
?? d_ave PICTURE "999.9" AT 18
a_ave = a_days / a_late
?? a_ave PICTURE "999.9" AT 29
n_ave = n_days / n_late
?? n_ave PICTURE "999.9" AT 39
ave_late = late_tot / late_delv
?? ave_late PICTURE "999.9" AT 48
?
WAIT
RETURN


*: EOF: RPT-7.PRG

-----

*:*************************************************************************
*:
*:        Program: C:\DATA\DBASE\RPT-8.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 07/29/93     14:46
*:
*:      Called by: BARPOP               (procedure in MAIN_MNU.PRG)
*:
*:           Uses: DWARFDB.DBF
*:               : BIDVEN.DBF
```

```
*:               : MUDKEY.DBF
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:*****************************************************************************
? "in report 8"
USE dwarfdb
USE bidven IN 2
USE mudkey IN 3
CLOSE DATABASES
RETURN
*: EOF: RPT-8.PRG

-----

*:*****************************************************************************
*:
*:        Program: C:\DATA\DBASE\RPT-9.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 07/29/93     14:46
*:
*:      Called by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:           Uses: DWARFDB.DBF
*:               : BIDVEN.DBF
*:               : MUDKEY.DBF
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:*****************************************************************************
? "in report 9"
USE dwarfdb
USE bidven IN 2
USE mudkey IN 3
CLOSE DATABASES
RETURN
*: EOF: RPT-9.PRG

-----

*:*****************************************************************************
*:
*:        Program: C:\DATA\DBASE\RPT-10.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 07/28/93     15:18
*:
*:      Called by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:           Uses: VRF-RATE.DBF
*:
*:        Indexes: VRF-RATE.NDX
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:*****************************************************************************
*  rpt-10.prg
*
*  program to list ABVM scoring trends (report 10)
*
*  28 jul 93
*
USE vrf-rate
ERASE vrf-rate.ndx
INDEX ON fsc_code+DTOC(score_date) TO vrf-rate
SET FIELDS TO fsc_code, overall_sc, score_date
GOTO TOP
?
? "FSC" AT 1, "Date" AT 9, "Score" AT 19
SCAN WHILE .NOT. EOF()
   IF score_date >= CTOD("02/01/91")
      ?
```

```
      ?? fsc_code PICTURE "xxxx" AT 1
      ?? score_date AT 8
      ?? overall_sc PICTURE "999.9" AT 20
   ENDIF
ENDSCAN
CLOSE DATABASES
RETURN

*: EOF: RPT-10.PRG

-----

*:**********************************************************************************
*:
*:        Program: C:\DATA\DBASE\RPT-11.PRG
*:
*:         System: ABVM Decision Support System
*:         Author:
*:      Copyright (c) 1993,
*:  Last modified: 07/29/93     14:47
*:
*:      Called by: BARPOP              (procedure in MAIN_MNU.PRG)
*:
*:           Uses: DWARFDB.DBF
*:               : VENDOR.DBF
*:               : OPENCON.DBF
*:
*:      Documented 08/12/93 at 15:17                SNAP!  version 5.01
*:**********************************************************************************
? "in report 11"
USE dwarfdb
USE vendor IN 2
USE opencon IN 3
CLOSE DATABASES
RETURN

*: EOF: RPT-11.PRG
```

Prototype ABVM DSS dBase IV Screens



Press number of menu choice, or highlight and press <Enter>

Report 1 - Summary ABVM Application Statistics

FSC	Number Awards	Percent ABVM	Percent w/ Diff	Average Diff ($)	Average Diff (pct)
1234	3	66.67	66.67	383.50	63.92
9876	2	100.00	50.00	1995.00	39.90
DSC	5	80.00	60.00	920.67	44.55

Press any key to continue...

Report 2 - ABVM Component Score Statistics

Part 2.1 - Awardee ABVM Statistics

total awards = 5

	number	overall average	delivery average	quality average
abvm awards	4	89.8	87.5	91.3
non-abvm awards	1	80.0	80.0	80.0
all awards	5	87.8	86.0	89.0

Press any key to continue...

Part 2.2 - Bidder ABVM Statistics

FSC	number awards	total bids	rated bids	average score bid	# ABVM awards	score ABVM bids	score non-ABVM bids	FSC ABVM score
1234	3	10	6	91.2	2	94.0	88.3	95.0
9876	2	5	2	88.0	2	88.0	0.0	87.0

Press any key to continue...

Report 3 - Quality Vendor Program Information

CAGE	Vendor Name	No. Awds	Award Dollars	Diff Awds	Diff Dollars	Bus Sz	Mf	Overall ABVM
00001	1234	1	1000.00	1	667.00			95.0
00001	9876	1	5000.00	1	1995.00			99.0
00001	abracadabra industri	2	6000.00	2	2662.00	s	y	98.0
00002	lipps inc	0	0.00	0	0.00	l	n	73.0
00003	1234	1	500.00	0	0.00			80.0
00003	intl business machin	1	500.00	0	0.00	s	y	86.0
00010	pittsburgh elephant	0	0.00	0	0.00		n	91.2
99999	9876	1	350.00	0	0.00			77.0
99999	racine industries	1	350.00	0	0.00	s	y	83.4

Press any key to continue...

Report 7 - ABVM Program Award Performance

FSC = 1234	ABVM Diff	ABVM No Diff	Non ABVM	Total
% delinquent	0.0	***.*	100.0	50.0
avg days late	***.*	***.*	14.0	14.0

Press any key to continue...

FSC = 9876	ABVM Diff	ABVM No Diff	Non ABVM	Total
% delinquent	***.*	***.*	***.*	***.*
avg days late	***.*	***.*	***.*	***.*

Press any key to continue...
Press any key to continue...

```
 FSC      Date       Score
 1234    02/01/91     73.0
 1234    02/15/91     78.0
 1234    03/01/91     88.0
 1234    03/15/91     83.0
 1234    04/01/91     89.0
 2000    02/01/91     75.0
 2000    03/01/91     77.0
 9876    06/01/93     87.0
Press any key to continue...
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1993	Final

4. TITLE AND SUBTITLE
Automated Best Value Model Decision Support System Functional Description

5. FUNDING NUMBERS

6. AUTHOR(S)
Kurt Schwarz

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
HQ Defense Logistics Agency (CAILR)
Cameron Station
Alexandria, VA 22304-6100

8. PERFORMING ORGANIZATION REPORT NUMBER
DLA-94-P20250

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ Defense Logistics Agency (AQP)
Cameron Station
Alexandria, VA 22304-6100

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Public release; unlimited distribution

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The DLA Operations Research Office (DORO) has developed a Decision Support System for tracking the usage and effectiveness of the Automated Best Value Model (ABVM). This Functional Description for the system details the type of information needed, the structure of the reports, and a mapping to the required data elements. The Functional Description can be used by DLA Pre-Award Contracting System programmers as the baseline for required system changes.

14. SUBJECT TERMS
Automated Best Value Model, ABVM, Decision Support System, procurement

15. NUMBER OF PAGES
95

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.